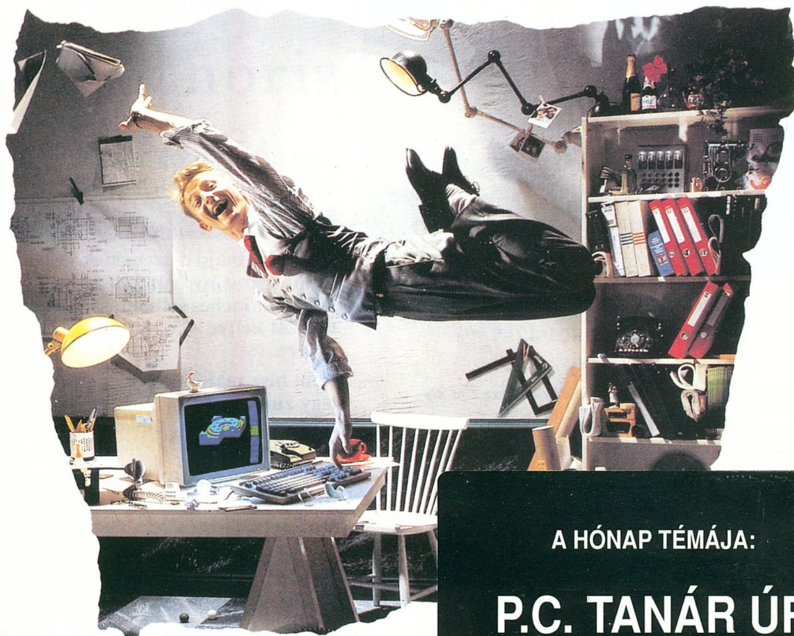
1991 / DECEMBER

ÁRA: 196 FT

ALAPLAP



MIKROSZÁMÍTÓGÉP MAGAZIN MÁGNESLEMEZ MELLÉKLETTEL



Az Alaplap alapkérdéseiről

A HÓNAP TÉMÁJA:

P.C. TANÁR ÚR

07C7 / 444543454D424552

PC–PAPÍR

AZ ALAPLAP VADHAJTÁSA

ÁRA: 1100100 FT

Karakterek vektortranszformációval

Érdemjegytőzsde iskolásoknak

RAM és ROM a romokon

Apróbb eLHAjlások

Menüzzünk!

A szoftver jogvédelméről

A MÁGNESLEMEZEN:

Csatlakozás a 87-es buszhoz
Ada, Modula, Pascal rutinok
Képernyővezérlő program
Sulibank
Játék: 5x5

Az Alaplap eddigi számainak tematikus tartalomjegyzéke

Vegye kezébe a minőséget!





Amit itt lát, azt Ön is kezébe veheti, és meggyőződhet róla, hogy milyen gyönyörűen nyomtat a Hewlett-Packard DeskJet 500-as, ha elmegy dealeréhez egy minta-nyomtatásért.

Nem hisz majd a szemének - a HP DeskJet 500-as tintasugaras nyomtató lézer minőségben nyomtat 300 pont/inches felbontással, nyomdai minőség hatását keltve - de mindössze egy mátrixnyomtató áráért.

Nem hisz majd a fülének - annyira halk, hogy a légy zümmögését is meghallja, miközben tökéletes minőségű levelek, táblázatok és grafikák készülnek el.

És megnyugtatásul: a Hewlett-Packard, mint a tintasugaras technológia feltalálója, a DeskJet 500-asra 3 év garanciát* vállal.

Tehát: ha meg kíván győződni arról, hogy a Hewlett-Packard DeskJet 500-as igazán jó vétel, kérjük látogassa meg a HP legközelebbi hivatalos dealerét és hozza el az alábbi szelvényt.

Az Ön HP Dealerét is gondosan választottuk és képeztük ki, így benne a legjobb partnerre talál, ha PC-t vagy perifériát kíván beszerezni.

* A garanciális javítás a HP szervizben történik.

ALBACOMP Kft. 8000 Székesfehérvár, Schönherz Z. u. 4/a
Telefon: (22)-27-532, Telefax: (22)-15-414

CONTROLL RT . 1091 Budapest, Üllői út 101.
Telefon: 114-0211, Telefax: 133-7392

DIGITAL Kft. 6723 Szeged, Csongrádi sgt. 83.
Telefon: (62)-56-530, Telefax: (62)-56-765

DUNA-ELEKTRONIKA RT. 1015 Budapest, Donáti u. 35-45.
Telefon: 201-7691, Telefax: 201-7773

EURO-CAL Kft. 6720 Szeged, Rákóczi u. 18.
Telefon: (62)-19-799, Telefax: (62)-19-799

INVENT-TRADE Kft. 4029 Debrecen, Ceglédi u. 4.
Telefon: (52)-15-580, Telefax: (52)-15-580

MIKRO-BIK Kft. 3527 Miskolc, Baross G. u. 13-15.
Telefon: (46)-53-100, Telefax: (46)-47-266

MICROSYSTEM RT. 1122 Budapest, Városmajor u. 74.
Telefon: 156-5366, Telefax: 155-92-96

R-COMP Kft. 1022 Budapest, Bimbó út 15.
Telefon: 135-9194, Telefax: 136-2250

- KQD -

Nem fogja elhinni, amíg nem látta a saját szemével.

SZELVÉNY

Vigyen el egy Ön által készített szöveget bármelyik hivatalos HP dealerhez. Dealerünk ki fogja azt nyomtatni Önnek egy DeskJet 500-ason.
Kérjük, adja meg az alábbi adatokat, és vigye el ezt a szelvényt a szöveget tartalmazó ASCII file-lal együtt valamelyik fent megadott dealerhez.*

Cég neve: ____________________

Az Ön neve: ____________________

Címe: ____________________

PC típusa: ____________ Hálózatban használja? igen nem

*) Ez a lehetőség 1991 december 31-ig áll fenn. Csak CP852-es kódkészletet használó PC-n létrehozott file-okat tudunk feldolgozni.
/ ALAPLAP 1991.December 13./



A VALÓRA VÁLT LEHETŐSÉG

A jólértesültek tudják, hová forduljanak:



HEWLETT
PACKARD

Authorized
Dealer

A Hewlett-Packard hivatalos dealerei Magyarországon:

ALBACOMP KFT.
8000. Székesfehérvár, Schönherz Z. u. 4/a
Telefon: (22)-27-532 Telefax: (22)-15-414

CONTROLL RT.
1091. Budapest, Üllői út 101.
Telefon: 114-0211 Telefax: 133-7392

DIGITAL KFT.
6723. Szeged, Csongrádi sgt. 83.
Telefon: (62)-56-530 Telefax: (62)-56-765

DUNA-ELEKTRONIKA RT.
1015. Budapest, Donáti u. 35-45.
Telefon: 201-7691 Telefax: 201-7773

EURO-CAL KFT.
6720. Szeged, Rákóczi u. 18.
Telefon: (62)-19-799 Telefax: (62)-19-799

INVENT-TRADE KFT.
4029. Debrecen, Ceglédi u. 4.
Telefon: (52)-15-580 Telefax: (52)-15-580

MIKRO-BIK KFT.
3527. Miskolc, Baross Gábor u. 13-15.
Telefon: (46)-53-100 Telefax: (46)-47-266

R-COMP KFT.
1022. Budapest, Bimbó út 15.
Telefon: 135-9194 Telefax: 136-2250

INFORMÁCIÓKÉRÉS: 09 ▶

Bárhol is látja ezt az emblémát, biztos lehet benne, hogy itt csak a legjobb minőségű termékeket és a legmagasabb színvonalú szolgáltatásokat kapja.

Függetlenül attól, hogy csúcstechnológiájú PC-kre, hálózati elemekre vagy perifériákra van-e szüksége.

A Hewlett-Packard lenyűgöző grafikákat létrehozó nyomtatói és a legfejlettebb CAD/CAM alkalmazásokhoz kifejlesztett plotterei új távlatokat nyitnak a felhasználók számára. A Hewlett-Packard az egyetlen gyártó, amely PC-hálózatokat és CAD/CAM eszközöket egyaránt szállít. Élvezze ennek előnyeit!

Keresse fel a Hewlett-Packard legközelebbi hivatalos dealerét, és ne felejtsen el tájékozódni páratlan garanciális szolgáltatásainkról is.

Hewlett-Packard: a legjobb választás lehetősége.



A VALÓRA VÁLT LEHETŐSÉG.

ALAPLAP

Mikroszámítógép magazin mágneslemez melléklettel

Megjelenik havonta

Főszerkesztő:
Faklen Pál

Főszerkesztő-helyettes:
Varga János

Szerkesztő:
Jakab Ágnes

Szerkesztőségi titkár:
Sziebig Andrea

A mágneslemez melléklet és a Közkincs szerkesztője:
Verebély Pálné

A Lemezkalauz szerkesztője:
Vékony Tamás

A szerkesztőbizottság tagjai:
Barna László
Boros György
Broczkó Péter
Brüll Károly
Farkas Ernő
Herczeg József
Kassay Árpád
Kónya László
Kovács P. Attila
Nagy Gábor
Pintér Gábor
Vargha Dénes
Villányi László
Zoltai Péter

Szerkesztőség, kiadó és hirdetésszervezés:
XI., Karolina út 17.
Budapest 1251
Telefon: 166-2111, 185-2192, 185-2421, 186-9644
Fax: 185-2221



Felelős kiadó:
Sebestyén Ilona igazgató
Cédrus Informatikai Rt.

Nyomdai előkészítés:
Tipoprint Kft., Budapest
Nyomtatás:
Zalai Nyomda, Zalaegerszeg
Felelős vezető: Galla József

Terjeszti a Magyar Posta. Előfizethető a hírlapkézbesítő postahivataloknál és a Posta Hírlapelőfizetési és Lapellátási Irodájánál (XIII., Lehel u. 10/a, Budapest 1900), vagy átutalással a 215-96162 pénzforgalmi számra.
Példányonkénti ár: 196 Ft
Évi előfizetési díj: 2 352 Ft

Külföldre terjeszti a Kultúra, Pf. 149, Budapest 1389

HU ISSN 0865-9788

A HÓNAP TÉMÁJA: P. C. TANÁR ÚR

SZÖVEGELŐ

SZERSZÁMOSLÁDA

KÖZELGÉP

MŰ-HELY

TEMATIKUS TARTALOMJEGYZÉK

SOLARSOFT LEMEZKALAUZ

Hazai műhelyekből

KÖZKINCS

ABAKUSZ

KIRAKAT

GÉPRAJZ

JOGÁSZ

FOGÓDZÓ

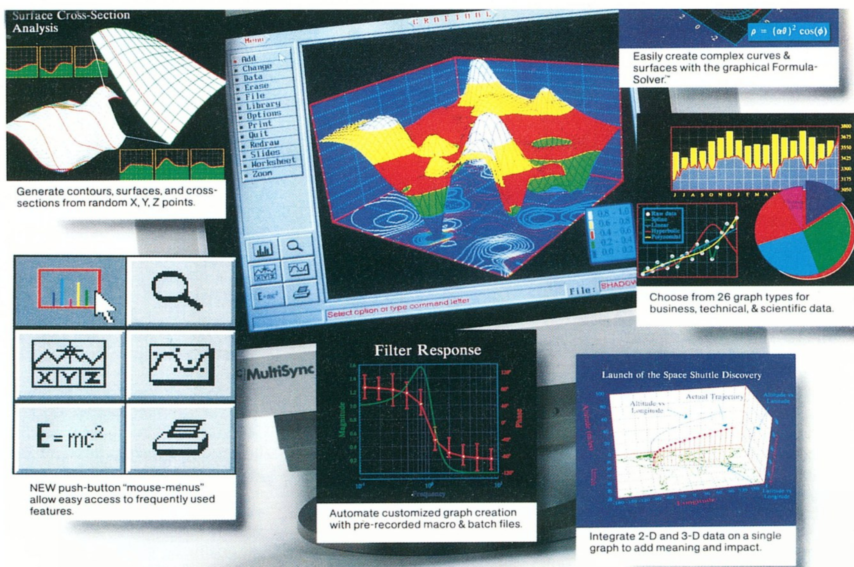
ALAPJÁRAT

PROGRAMOZÁSTECHNIKA

VISSZACSATOLÁS

MÁGNESLEMEZ MELLÉKLET ❑

Címlapképünkön a „Happy User" GmbH reklámfotója



Tanulni, okulni, tanítani

Az élet sokszínű, mindig is az volt, csak nem annyira, mint most. Feltételei változékonyak, mindig is azok voltak, csak nem annyira, mint most. Az emberiség nem, noha az egyes ember mindig is választhatott, hogy úrrá legyen-e a változékony feltételeken vagy azok áldozatává váljék. A mai homo sapienstől — de még inkább a holnapitól — idegen az áldozattá válás, a lemaradás, az életfeltételek változékonyságára való felkészületlenség gondolata.

Semmi nem tükrözi ezt jobban, mint az alig néhány év alatt világszerte százmilliókat mozgósító, szüntelen felkészülésre, tanulásra serkentő, forradalmian új ismereteket, készségeket és egyben életlehetőségeket kínáló informatika, s azon belül is a számítástechnika. Ma már természetes dolog, hogy az ember felnőtt korban is folyamatosan tanul, azonban a következő generációnak az élet első lépéseire való felkészítése, az ahhoz szükséges ismeretekkel való ellátása változatlanul az iskolák alapvető hivatása, kötelessége. A számítástechnika (elsősorban a programozás és az alkalmazás) oktatása örvendetes módon hazánkban is egyre szélesebb körűvé válik.

Az iskolák azonban alap-, középvagy felsőfokon sem mindig és mindenütt rendelkeznek kielégítő adottságokkal, eszközökkel ahhoz, hogy vállalhassák a számítástechnikai ismeretközlés egységes mederbe terelését, megfelelő tananyagokkal és kiérlelt módszertani segédletekkel való ellátását. Ugyancsak elodázhatatlan problémának tűnik a kellő számú és felkészültségű számítástechnikai szaktanár központilag szervezett kiképzése, de még a bármilyen fokon egységes tananyag, tankönyv kiadása sem reális elképzelés pillanatnyilag.

Ezt támasztja alá a számítástechnika sokrétűsége; az iskolák eltérő számítástechnikai felszereltsége; a szaktanárok, gyakorlati oktatók sokféle, de más-más koncepciójú készenléte az ismeretek átadására; az egyes intézmények jellegéből vagy környezetéből származó, képzésorientáló igényeinek különbözősége; a számítástechnikai hardver- és szoftvereszközök típusainak, struktúrájának, verzióinak gyors változása.

Már csak az elmondottak alapján is megoldhatatlan akár egyetlen iskolafokozatra illesztve az optimális tananyag összeállítása, hiszen az „optimum" iskolánként más és más lehet.

Merre van előre? Hogyan válasszon a szülő — később maga a diák, utóbb a már dolgozó ember —, ha a jövőre gondol? Ki és miért érezzen/érezhet felelősséget; s mit tegyen/tehet, hogy nyugodt lelkiismerettel nézzen évezredünk végének követelményei elé? E havi összeállításunk ebben a vonatkozásban kíván támpontokat adni — nézetünk szerint kivétel nélkül érdekelt — olvasóinknak.

Számítástechnikai oktatás a közoktatási intézményekben

Általánosan, de nem közepesen...

Az általános és a középiskolák tanulói számára iskolai keretek között többféle lehetőség nyílik a számítástechnikai, az informatikai és a személyi számítógépek használatához szükséges ismeretek elsajátítására.
Pedagóguson és tanulón egyaránt sok múlik, hogyan is élnek ezekkel.

A legtöbben a szakkörökben ismerkednek a számítógéppel. Az elmúlt, azaz az 1990/91-es tanévben például az általános iskolai tanulók közül 37 300 fő 2666 szakköri csoportban foglalkozott számítástechnikával. A középiskolások számára kínált 1045 szakkört 15 700-an, a szakmunkástanulók részére szervezett 609 csoportot pedig 9500-an látogatták.

Az egyik legnépszerűbb tárgy a számítástechnika/informatika a fakultatív oktatásban is. Az általános iskolák közel 1/3-ában működtek ilyen csoportok, és ezekbe összesen 23 700 hetedik és nyolcadik osztályos tanuló járt. A gimnáziumi tanulók közül ugyanekkor kereken 4400 fő választotta fakultatív tárgyként a számítástechnikát. Évről évre több száz gimnazista a számítógép-kezelői ismeretekből érettségi vizsgát is tesz.

Nincsen szakma gépek nélkül

A számítástechnika önálló tantárgyként vagy valamely szakmai tantárgyba beépítve ma már szinte mindegyik középfokon oktatott szakmának az ágazati nevelési-oktatási tervében szerepel. Kisebb számban vannak számítástechnikára szakosodott osztályok alapfokon is. Számos példa van továbbá a számítástechnika oktatási eszközként való használatára is.

Az állami (önkormányzati) iskolákban a tanulók középfokú számítástechnikai szakképesítést egyrészt a közgazdasági, másrészt a műszaki szakképzés keretében szerezhetnek. Az 1990. októberi statisztikai adatok szerint kereken 4900 fő vett részt ezekben a képzésekben. Szakmánként, ágazatonként a létszámok a következőképpen alakultak:

- számítástechnikai programozó: 2377 fő
- számítástechnikai folyamatszervező: 667 fő
- számítástechnikai programozó, képesített könyvelő: 171 fő
- számítástechnikai műszerész: 673 fő
- információ- és számítástechnikai technikus: 485 fő
- gépgyártás-technológiai, számítástechnológiai technikus: 503 fő

A jelzett szakmákra felkészítő állami (önkormányzati) intézmények címlistáját a mágneslemezen, mellékletként közöljük.

Középiskolás fokon

A szakképzésen kívül a gimnáziumi tanulók közül egyedi tanterv szerint 414-en tanultak számítástechnikai ismereteket. A fakultációk keretében 2163 fő számítógép-kezelői ismereteket és 959 fő pedig programozást tanult.

A számítástechnika gyors fejlődését csak a tananyag, a tanulóknak nyújtott ismeretek és követelmények, valamint az eszközbázis állandó megújításával lehet követni. Ezzel van összefüggésben az a változás is, hogy napjainkban az IBM, illetve az IBM-kompatibilis PC gépek elterjedésével az iskolákban fokozatosan háttérbe szorul a különböző számítógépes nyelvek oktatása, és helyette az egységes informatikai megközelítés kerül.

A szakképzésben is csak a fő irányok vannak kijelölve. A közgazdasági szakközépiskolákban a számítástechnikai programozói és a folyamatszervezői szakképzés célja: közgazdasági szemlélettel, ismerettel rendelkező középszintű számítástechnikai szakemberek képzése. A tanulók 4 év alatt megismerik a számítástechnikai munkakörökben jelentkező gazdasági feladatok megoldásának szervezési módszereit. Képesek lesznek programok tervezésére, készítésére, valamint kész programok alkalmazására.

A programozó és könyvelői képesítéssel a tanulók a számviteli szemléletű szaktudás mellett programozói és PC-s könyvelői ismereteket szereznek. Mindhárom ágazatban magas óraszámban oktatnak angol nyelvet, s az átlagnál több a matematikai órák száma is.

A számítástechnikai műszerészek a számítógépek gyártásával, karbantartásával kapcsolatos alapvető feladatok ellátására kapnak felkészítést. A két technikus szak pedig a villamosipari, illetve a gépipari alkalmazási szakfeladatok megoldásához, szakmai programok alkalmazásához szükséges ismereteket nyújt.

Szerte az országban számos kísérlet van folyamatban a számítástechnikai oktatás tartalmi megújítására, az informatikai eszközök alkalmazásának bővítésére. Ezek a kísérletek főleg az oktatás feltételeinek javítására és új programok kidolgozására irányulnak. A kísérleti programok között a más tárgyak oktatását segítő és a számítástechnikai tárgyak tartalmát megújító kezdeményezések egyaránt vannak. Az ered-

> Az általános és a középiskolák számítástechnikai fejlesztéséhez jelentős mértékben járult hozzá az 1986-1990-es években az EGP (az Elektronizáció Gazdaságfejlesztési Programja) közoktatási részprogramja, amelynek keretében az iskolák több mint 170 millió forintnyi támogatást kaptak a Művelődési Minisztériumtól, s ennek kb. a kétszeresét a tanácsoktól.

Számítástechnikával kapcsolatos mutatók

Iskolatípus	Számítógépek	Egy gépre jutó tanulók	Vásárolt szoftver 1000 Ft-ban	Egy gépre jutó vásárolt szoftver Ft-ban	Egy tanulóra jutó vásárolt szoftver Ft-ban
Általános isk.	20 260	55	24 596	1214	21,8
Szakmunkásképző	3562	59	10 280	2886	49,1
Középiskola	12 602	23	32 254	2559	110,5
Összesen	36 784	44	67 130	1825	41,1

mények terjesztése, hasznosítása is szerves része a fejlesztő munkának.

Ahol és akik belevágtak

A közoktatási innovációt segítő Közoktatásfejlesztési Alap 1991. évi pályázatai alapján ismert, hogy a számítástechnikai szakképzés megújításán az egri Neumann János Szakközépiskola pedagógusai dolgoznak.

Békéscsabán és Oroszlányban a számítástechnika általános iskolai szakosított tantervének kimunkálásán dolgoznak — az itt szerzett tapasztalatok továbbfejlesztésével.

A számítástechnikának az általános iskola kezdő szakaszában, az 1-3. osztályokban való alkalmazásával kapcsolatos módszerek kidolgozása folyik Nyíregyházán, Debrecenben, Tatabányán, Maglódon, valamint Budapesten

Az első XT és AT típusú személyi számítógépek a 80-as évek közepén jelentek meg a közoktatási intézményekben, s azóta gyors ütemű volt a térhódításuk. A középiskolákban a számítástechnikai oktatás ma már elsősorban PC gépekre épül. Az alapfokú iskolákban azonban közel sem volt ilyen látványos a fejlődés, itt még mindig a COMMODORE kategória van túlsúlyban. Tény az is, hogy az elmúlt tanévben még 700 olyan iskola volt — elsősorban a távoli, apró iskolák közül — ahová még a COMMODORE típusú kisgépek sem jutottak el.

A számítástechnika, informatika tantárgy bevezetése kísérleti tanterv szerint zajlik két szegedi, két fővárosi iskolában, valamint Debrecenben, Hajdúszováton, Galgamácsán, Tatabányán, Oroszlányban, Karmacson és Kápolnásnyéken egy-egy iskolában.

a VII. és a XIX. kerületben. (Maglódon már az óvodában is vannak játékos számítógépes foglalkozások.) A LOGO program oktatásával kapcsolatosan három helyen is gyűlnek a tapasztalatok. Nyíregyháza egyik iskolájában a 3.-4. osztályosoknak a magyar nyelv és a matematika tanításához, Győrújbarátiban pedig a fizika tanításához használják a személyi számítógépet.

Szegeden a siketek anyanyelvi nevelésében és a szaktárgyak oktatásában segít a számítógép.

A 80-as évek végéig az állami iskolák számítástechnika-oktatási feltételeinek megteremtését a költségvetés terhére finanszírozott programok és források segítették. A számítástechnikai célú központi fejlesztési források azonban mára nagyrészt megszűntek. Állami támogatást napjainkban az oktatási intézmények csak a Közoktatásfejlesztési Alaptól és a Szakképzési Alaptól kérhetnek. A fejlesztések, kísérletek forrásainak megteremtésében egyre nagyobb szerepet kapnak az e célra létrehozott vagy más alapítványok.

Amire érdemes áldozni

A számítástechnikai eszközök és ismeretek alkalmazásának fontos helye van az újonnan alakult alapítványi szakképző intézményekben is. Jó példaként szolgál erre az 1991 szeptemberében indult Közgazdasági Politechnikum programja. Az Állami Biztosító Rt., az Iskolapolgár Közéleti Alapítvány, a Magyar Hitel Bank Rt., valamint további 14 szervezet, illetve alapítvány, alap által támogatott Közgazdasági Politechnikum a képzés egészét átfogó vállalkozói szemléletet és szakértelmet közvetítő programot kíván megvalósítani. A modern piacgazdaság számára átfogó általános műveltségű és biztos nyelvtudású, kockázatvállaló, döntésképes, szakmájához professzionális szinten értő, és a vállalkozás gazdasági tevékenységében is jártas, tulajdonosi szemléletű szakemberek képzését tűzi ki célul. Felismerve azt, hogy mindezekhez informatikai tudás is kell, a számítástechnikát és informatikát szervesen beépítették a tananyagba az alapképzés tárgyai közé.

Programjuk szerint a számítástechnika és informatika tantárgy oktatásával nem informatikusokat vagy számítástechnikusokat kívánnak képezni, hanem olyan közgazdászokat, akik szívesen és hatékonyan dolgoznak az információtechnikai eszközökkel. El szeretnék érni, hogy a tanulók legyenek tisztában az információtechnikai eszközök határaival, korlátaival, és a problémáikat informatikai, számítástechnikai nyelven is meg tudják fogalmazni.

A számok beszélnek...

A számítástechnikai, informatikai oktatást nagyban befolyásolják az eszközfeltételek. A Művelődési és Közoktatási Minisztérium statisztikai osztályának gondozásában a művelődési ágazat számítástechnikai eszközellátottságáról szóló statisztikai tájékoztató a közoktatási intézmények 1990. évi helyzetéről a fenti adatokat közli. Tudni lehet a hivatkozott tájékoztatóból, hogy az iskolák számítástechnikai eszközökkel való ellátottságát még korántsem lehet teljesnek tekinteni.

Szalay Lásznóné — Práff Csaba

Nemzeti alaptanterv informatikából

Forrongás a „forrásközpont” körül

Ez év áprilisában készült el az Informatika Nemzeti Tantervének második fogalmazványa. A szerkesztésben közismert általános és középiskolai, főiskolai és egyetemi oktatók vettek részt: Borbola István, Ballér Endre, Báthori Zoltán, Nagy József, Szebenyi Péter, Lukátsy Sarolta, Makay Árpád, Perge Imre, Selényi Endre, Sikala János, Szűcs Ervin, Theisz György, Varga László. Ez részben garancia is lehetne arra, hogy ez a tanterv az elmúlt negyven évben alkotottakkal ellentétben jól átgondolt, precíz és használható lesz.

Az eddigi időszakban a tantervek kidolgozásánál nem volt mód — vagy nem akarták a szerkesztők — a nyilvánosság bevonására a munkába, most viszont az a cél, hogy az Informatika Nemzeti Alaptantervének elkészítésénél minél több szempontot, javaslatot és bírálatot vehessenek figyelembe a jó végeredmény érdekében. Ezenkívül fontosnak tartom, hogy ne csak a tanárok, és közülük is csupán azok a tanárok vegyenek részt a szerkesztésben, akik csak „kisiskolás” korukban láttak általános és középiskolát, hanem azok is tehessenek javaslatot, akik már évek óta ezekben az intézményekben oktatnak. Az eddig elkészült rossz tanterveknek az is súlyos hibájuk volt, hogy összeállításuknál ha meg is kérdeztek aktív tanárokat, akkor azok többsége a főiskolák és az egyetemek gyakorlóiskoláiban tanított, ahová köztudottan protekciós, tehetséges gyerekek kerülhettek be. A rajtuk kipróbált oktatási módot és követelményszintet átlagos iskolákban nem lehetett tartani.

Én a vitaanyag olvasásakor találtam benne üdvös és kevésbé tetsző gondolatokat, továbbá olyan ötletet is, amelyet alapjában véve jónak tartok, de a megvalósítása nagyon fáradságos és nehéz munkát ró a tanárokra.

Az alaptanterv vázlatos ismertetésével egy általános vitát szeretnék kiváltani, ami nem azt jelenti, hogy én nagyon sok pontban nem értek egyet vele. Szándékom csupán annyi, hogy felhívjam mindenkinek a figyelmét erre a tervezetre, amely évekre meghatározhatja a fejlődést, meghatározza a tanárok feladatát. Nemcsak a számítástechnikát oktatókét, hanem az egész tanári karét, ugyanis az informatika tanításának fő koncepciója: ne egy tantárgy keretében szerezzék meg a gyerekek ezeket az ismereteket, hanem általánosan — elrejtve a többi tárgy közé. Ezért ez más tárgyak tantervével ellentétben sokkal átfogóbb, egyben módosítja azokat is; több tanárt érint — ezért úgy gondolom, hogy valóban nagyobb nyilvánosságot is igényel.

Mielőtt rátérnék az alaptanterv legfontosabb részeinek az ismertetésére, fontos tisztázni, hogy mit nevezünk informatikának. Sokan azt gondolják, hogy ez a számítástechnika újabb elnevezése, sokan abban a hitben vannak, miszerint ez a számítástechnika és a kibernetika összefoglaló neve. Ezért számos helyen ezt a tantárgyat a számítástechnika-tanárokra bízzák: tanítsa meg ő az informatika alapfogalmait. Ez nem helyes így. A számítástechnika-tanárok vállalják csak a számítástechnika oktatását, ami része az informatikának is, de ez utóbbi sokkal tágabb értelmezésű.

Hogy az alábbiakban olvasható követelményrendszerű tantárgyat miként tudják a tanárok maguk között szétosztani, kemény dió. Ezért vita tárgya az is, hogy ha jó is a koncepció, azt hogyan lehet megvalósítani. (Senkit nem kívánok a vitaanyag elbírálásának objektivitásában befolyásolni, ezért itt és most szeretnék semleges maradni: sem a részemről jónak, sem az éppenséggel rossznak tartott részekhez nem fűzök kommentárt.)

Az informatika elnevezést hiába kerestem a Magyar értelmező kéziszótárban, a Számítástechnikai kislexikonban, nem találtam. (Új keletű lenne a kifejezés?) Az Idegen szavak és kifejezések szótára a következőket írja:

„Információ — értesülés, adat, hír, tájékoztatási anyag; a kibernetika elméletében: az anyag tulajdonságainak visszatükröződése jel formájában.”

„Informatika: az információk megszerzésével, tárolásával és feldolgozásával összefüggő ismeretek összessége.”

Most nézzük, hogyan értelmezik —kissé bonyolult megfogalmazásban — a szerkesztők ezeket a fogalmakat:

„Az információ olyan hír (közlés), amely valamilyen bizonytalanságot szüntet meg. Ezért az információmennyiség növekedése az adott rendszer határozatlanságának csökkenésével jár. A rendszerek kölcsönhatásukkor információt is cserélnek. Így az információ — tőlünk függetlenül — velejárója a legelemibb természeti és a legmagasabb szellemi, társadalmi folyamatoknak. Napjaink technológiája hatalmas mennyiségű információt zúdít az emberekre. A hatékony információkezelés ezért mindenkinek egyaránt feladata és érdeke.”

„Az informatika önálló tudomány. A tárgykörébe tartozó problémák pedig a tudományágak között többvetületűek. Éppen ezért az egyes tudományágak képviselői — szakterületük szerint — más és más helyre szokták tenni a hangsúlyt az informatika meghatározásában. E hangsúlyoktól függetlenül mára az informatikát a világban mindenhol a szisztematikus információkezelés tudományának tekintik. Az informatika ennek az információkezelési folyamatnak a központjába az alkotó embert állítja. Ezért az általános képzés során az informatika műveltségterületében a szisztematikus problémamegoldási folyamathoz kapcsolódó, általánosan alkalmazható módszerek, az automatikus és (kész) problémamegoldó eszközök, berendezések és a forrásközpont felhasználási képességének kialakítása

kapnak nagy szerepet. Háttérbe szorul az információelmélet, a rendszerelmélet, a speciális alkalmazások köre, mert ezek egyes elemeit az általános képzés szintjein az érintett szaktárgynak célszerű megemlítenie."

A szerkesztők felhívják a figyelmet arra, hogy az alaptanterv nem a régi értelemben vett tanterv. Jellemzői a következők:

1. Az alaptanterv nem tantárgyakat, hanem műveltségi területeket tartalmaz. Ennek alapján az iskola pedagógusai állítják össze a saját iskolájuk igényeihez és szükségleteihez illeszkedő tantárgyi rendszert. Ehhez a munkához segítséget jelent a pedagógiai programok, tankönyvek, taneszközök piaca, kínálata.

2. A tananyagot az alaptanterv nem évfolyamok szerint, hanem két szinten fejti ki. Ebből következően az évfolyamonkénti óratervet is az iskolában dolgozzák ki.

3. Az elemi szint az első hat évfolyam, az alapszint a további négy évfolyam anyagát tartalmazza.

Hogyan épüljön be az „Informatika" a helyi tantervekbe?

A Bevezetőből megtudjuk, hogy az ismeretszerzés, a kvalifikált munka, a szórakozás elképzelhetetlen könyvek, folyóiratok, audiovizuális dokumentumok és a modern elektronikán alpuló informatikai, kommunikációs technikai eszközök, módszerek és eljárások nélkül. Itt olvashatjuk a következőket is:

„Az embert körülvevő rendszerek és folyamatok információtömegének gyors és rendszerezett kezelése jelenleg elsősorban elektronikai berendezésekkel, számítógépekkel biztosítható. Ezért a hagyományos és az új technológiákon alapuló informatikai eszközök és módszerek felhasználási készsége olyan szükséges része az egyének műveltségének, amelyet már az általános képzésben el kell sajátítani. Az általános képzés minden műveltségterületén az adott terület sajátosságainak megfelelően meg kell jelennie az informatikai szemléletnek, az eszközöknek és a módszereknek. A NAT (Nemzeti Alap-Tanterv) Informatika nem a tudományág kronológiája és definíciórendszere alapján épül fel, hanem a problémához szükséges alkalmazások szempontjából előnyös csoportosítás szerint."

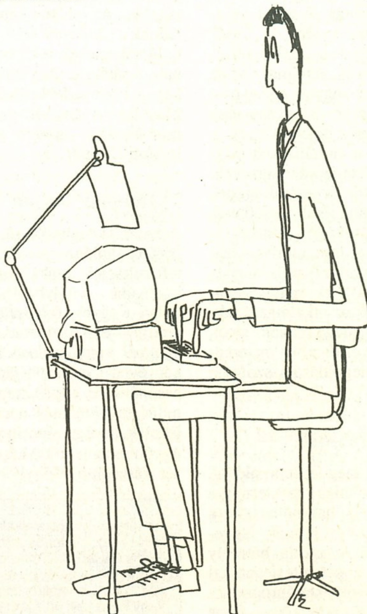
(Az alaptantervet részletesebben a mágneslemez mellékletben ismertetjük.)

A tapasztalatok azt mutatják, hogy már kisgyermekkorban sikeresen kialakítható az informatikai eszközök felhasználási képessége. A gyerekek éppen úgy a környezetük részének tekintik például a számítógépet vagy a videomagnót, mint a munkafüzeteket vagy a könyveket. Nem a tanulók életkora, hanem az adott környezet feltételei (szakképzett nevelő, megfelelő eszközök stb.) határozzák meg, hogy az informatikai készségeket, képességeket, attitűdöket a tantárgyak rendszerében elosztva vagy önálló tantárgy segítségével alakítja-e ki az iskola. Az informatika nem számítástechnika, de ez nem jelenti azt, hogy az informatikai műveltség jelentős részét ne lehetne egy jó tematikájú „Számítástechnika" tantárgy keretében elsajátíttatni. Ez a számítástechnika azonban nem lehet az általános képzésben programozásoktatás. Még inkább elkerülendő a hagyományos Basic tanítása. De ez sem jelenti azt, hogy ott, ahol adottak a feltételek, a tanulók egy érdeklődő része ne ismerhetne meg és ne használhatna valamilyen programozási nyelvet problémájának megoldásához.

Az informatika nem technika, de rendkívül sok ilyen vetülete van. Az információtechnológiai berendezések technikai eszközök, s a „külvilággal" (az emberrel is) elektromechanikus, elektronikus úton tartják a kapcsolatot. Az egész információkezelési folyamat technikai rendszereken és eszközökön alapul. Éppen ezért a „technika" tantárgy is alkalmas az informatikai műveltség jelentős részének a közvetítésére. Az általános képzésben az informatika technikai megközelítése nem a rendszerelmélet kialakulását, hanem az eszközök megfelelő szintű megismerését és felhasználási képességét kell, hogy eredményezze. Az életkori sajátosságok miatt az informatika technikai megközelítésével elsősorban a speciális érdeklődésű tanulókra érdemes számítani.

Az „Informatika" alaptanterv követelményrendszerében csak olyan témakörök szerepelnek, amelyek önálló alapozó tantárgyként is taníthatók. Az általános képzés utolsó szakaszában (13-14 éves kortól) ennek a tantárgynak a bevezetését javasoljuk. Természetesen ahol a feltételek adottak, ott a tantárgy korábban is indítható.

Kovács P. Attila



Lineáris operátor

Kréta — vagy PC?

Békés győzelmecskék Békéscsabán

A számítástechnika oktatásának hagyományairól nem beszélhetünk, inkább kísérletezésekről, próbálkozásokról, itt-ott bevált metódusokról. A szakmai viták napjainkban is javában folynak; van, aki aktívan részt vesz ezekben a vitákban, s akad, aki szerint: vitatkozzatok csak, én dolgozom. Jómagam vitázni is hajlamos vagyok, de... Elsősorban saját iskolámban szerzett tapasztalataimat szeretném közreadni — a módszerek, amelyeket közel négy éve alkalmazunk, nálunk beváltak.

A gépet már az első osztályba célszerű bevinni. A tanulók matematikaórán a „NYUSZI" programcsomagot (amely remélhetőleg a közeljövőben megjelenik a SolarSoft programkönyvtárban is) használják, magyarórákon az olvasást, elválasztást, helyesírást gyakorolják. Később ez kiegészül a környezetismereti és nyelvi órákon való számítógépezéssel. A programok PC-s változatai rögzítik a tanuló nevét, teljesítményét, a dátumot, időt. A felsős osztályok már történelem-, fizika-, földrajz-, kémiaórákon is alkalmazzák az oktatóprogramokat; vagy az egész osztály, vagy annak csak egy része vesz részt. Ezek az órák egyúttal arra is alkalmasak, hogy minimális számítástechnikai ismereteket is közvetítsenek.

Sokan úgy vélik, a számítógépek elsősorban matematikaórákra való eszközök. Pedig a számítógép minden tanárnak a kezében az lesz, amilyen szakos, és azt fogja tükrözni, amit maga is fontosnak tart. Ha csak matematikaórákra korlátozódik az alkalmazás, akkor a gyermeket meggyőzzük arról, hogy a számítógép elsősorban matematikai problémák megoldására szolgáló eszköz.

„Csempész"-stratégia

Minden iskolában meg kellene oldani, hogy a számítástechnikai szaktermet a tanulók önállóan is használhassák a tanuláshoz. Mi ezt úgy hívjuk, „ügyelet". A 7. órában az iskola bármely tanulója bejöhet a számítástechnikai terembe, és bármelyik oktatóprogramot kérheti, „nyúzhatja". A tankönyv, füzet mellett a témához kapcsolódó oktatóprogramok érdekesebbé, így hatékonyabbá teszik a tanulást. Ehhez elsősorban az iskolavezetés támogató szemléletmódja szükséges; jó oktatóprogramok és legalább egyetlen olyan tanár, aki felismeri ennek igazságát.

A folytatás sikeres lehet, ha a napközis csoportok délutánonként rendszeresen igénybe veszik az oktatóterem szolgáltatásait. (Itt elsősorban az oktatóprogramokkal végezhető gyakorlásra gondolok.) Így már annyira kihasználható egy szaktanterem, hogy ez meggyőző érv lehet bármely önkormányzat számára.

Továbbá: a felzárkóztató órák, a teremben sokkal eredményesebbek, mint hagyományos formában. Egyszerűen azért, mert a számítógép olyan médiaeszköz egyben, amellyel igen kevés pedagógusegyéniség képes felvenni a harcot a figyelem felkeltéséért.

A játék a 7-14 éveseknek legalább olyan fontos, mint a tanár szerint a tanulás. Az ehhez kapcsolódó forma nálunk a „Mikro-Klub" fedőnevet viseli. Amikor ezek a klubtalálkozók vannak, a játék, a programcsere zöld utat kap. A könnyedebbnek látszó foglalkozásokhoz a tanulók szélesebb tábora csatlakozik — s nem is sejti, hogy itt is tömik a fejét...

Egyenetlen output

A számítástechnikával való ismerkedést az általános iskolákban elsősorban a fakultációk, szakkörök biztosítják. A nyolcadik osztályból kikerült tanulók olyan vegyes felkészülést mutatnak, amilyen elképzelhetetlen más tantárgyaknál vagy szakköröknél. Van, aki két éven keresztül játékprogramokat használt, de a program betöltésén és indításán kívül mást nem ismer. Vannak iskolák, ahol az algoritmusszerkesztést, a problémamegoldó képesség fejlesztését részesítik előnyben, programokat készítenek: számítástechnikai ismereteik „megalapozottak".

Egyes iskolák informatikai vagy számítástechnikai szakosított tantervű oktatásra vállalkoznak, kísérleti jelleggel. Ezek többségét a Közoktatás Fejlesztési Alap karolta fel, és biztosítja a minimális anyagi hátteret. Ha a KFA megszűnne, ezek a kísérletek is elhalnának, pedig egy-egy lehetőséget hordoznak, hogy ne csupán eseti, hanem egységes, mérhető ismeretanyaggal rendelkezzenek a tanulók. Ilyen például a Farkas Károly által elkezdett kísérlet, mely az általános iskolák alsó osztályainak programja, és több iskola is átvette már. Vagy: a saját iskolámban három évvel ezelőtt elkezdett, szakosított tantervű, hatéves program, melynek felhasználásával az 1991-92-es tanévben közel 300 kisdiák kezdte el az ismerkedést a számítástechnikával.

> A számítástechnika alkalmazásának terjedését az általános iskolákban erősen lefékezi al fenntartásukért folytatott, mindennapos küzdelmek, s ahol az a kérdés, hogy a tanítás folytatásához PC vagy kréta kell, ott a krétát fogják választani.

„Helpek" nélkül

Vágyálomnak tűnhet, hogy heterogén géppark mellett egységes tematika, egységes tankönyv alapján oktassanak az intézmények. Olyan programok, amelyeknek a felhasználói felülete megegyezik mind a Commodore, TVC, PC gépeken, gyakorlatilag nincsenek, vagy csak elvétve. Hogy ilyen programokra nagy szükség lenne, azt gondolom, nem vitatja senki. Nem lehet úgy oktatási intézményeket magukra hagyni, hogy vannak ugyan eszközeik, de sem a gyártók, sem a forgalmazók nem vesznek tudomást arról, hogy ezek milyenek — egyszerűen azért, mert ez nem üzlet számukra. Holott sem az alapfokú, sem a középfokú oktatási intézmények működése nem helyezhető üzleti alapokra. Példaként: a mágneslemezen van egy olyan programnyelv, amelyet 9 éves kortól használunk a szakosított tantervű oktatáshoz. Ehhez tanterv, tanmenet, tankönyv, tanári segédlet, a tematikát támogató programok egységesen megtalálhatók mind C+4, mind TVC és PC gépekre. A modulok, az alapvető gépkezelés sajátosságait tartalmazzák géptípusonként, de a tematika megvalósításához már egységes tankönyv áll rendelkezésre.

Schneider Ferenc

Programozni vagy nem programozni...

Létkérdés vagy logikai feladvány?

Előre leszögezem:
nem a szakmai képzésről kívánok szólni!
Egy számítástechnikai
— de csak egy számítástechnikai —
szakképző intézményben egészen mások
a felvetendő kérdések.
Ez az eszmefuttatás azokról a középiskolákról szól,
ahol az általánosan művelt,
akár közép-, akár felsőfokon
szakmát tanulni képes embereket nevelik.

Egy biztos: ma az általános műveltségnek mindinkább részévé válik a környezet megfelelő ismerete, kezelése. A „természetellenes" környezet valóban sok-sok kárt okoz, de lemondani sem tudunk róla. Divat azonban nem érteni a technikához (mosolyogtató sztorikkal mindahányan találkoztunk már...).

A számítástechnikai szaktanár azt gondolhatja: „ha átadom a tudásom egy (kisebb-nagyobb) részét, felkészítem a gyereket. Jobb esélyei lesznek az életben." Így hát elkezdi úgy, ahogy őt tanították a nagygépeken felnőtt öregebbek: a gép felépítése, a belső adatmozgások, majd egy magas szintű nyelv a kívánatos alapossággal — struktúrák, algoritmusok stb. —, és csak ezután tér rá az alkalmazásokra. Még szerencse, ha nem öli tanulóit a Boole-algebrával vagy a tizenhatos számrendszer szabályaival!

A középiskolai számítástechnika-tanárok feladata azonban nem ez. A számítástechnika szakma, és a programterv legalább annyira mű, mint egy épületé! Lehetnek persze érdeklődő diákjaink, akik — például szakkör keretében — megismerkednek e szakmával, mint ahogy iskolaújságot is írnak tanulóink...

„Éhe az újnak"

Cél a szemlélet kialakítása, olyan emberek nevelése, akik nem menekülnek sikítva az újdonságoktól, akik nem remegnek, ha ezentúl valami megszokott dolgot máshogy, mással kell elvégezni. Ezt hívják konvertálható tudásnak. Számítástechnikából ez igen egyszerű feltételrendszert eredményez!

1. Nem programozókat, hanem felhasználókat képzünk! (Akár ne is tudja meg a gyerek, hogy mi az a bájt. De ismernie kell, hogy milyen belső logikával, hogyan viselkedik a gép, a háttértároló. Tudja használni az operációs rendszert — de nem kell azt programoznia is. Értse meg a programot, lássa meg a hiányosságait, de nem szükséges előállítania.)

2. Nem programot, hanem alkalmazást kell tanítani! (Ma már számtalan felhasználási terület van. Ha ezeket jobban megnézzük, rájövünk: ez a sok-sok terület nem is olyan rengeteg, különösen, ha a speciális szakmai tudást kívánó alkalmazásokat kizárjuk a vizsgálatból — például a CAD-et. Így értelmes, de nem csakis a szórakoztatást célzó tevékenységnek a szöveg- és adatfeldolgozás, valamint a kommunikáció marad. Ezt egészíti ki a rendszer működtetéséhez szükséges ismerethalmaz, azaz a megfelelő operációs rendszer és segédprogramjai — például a shellek. Tudom, hogy ez durva egyszerűsítésnek tűnhet, hiányolhatják egyesek például a grafikai alkalmazásokat. Csakhogy: ezek a programok vagy kivételes igényeket elégítenek ki, és különleges szaktudást követelnek — ilyen a már említett CAD vagy a reklámgrafika —, vagy mégiscsak a játék szintjén maradnak.)

Hogy alkalmazást tanítunk, az én felfogásomban ezt jelenti: például a tanulót nem az általunk legjobbnak tartott szövegszerkesztő használatára kell megtanítani, hanem arra, milyen műveletek végezhetők el egy standard szövegfeldolgozó programmal.

3. Nem elméletet tanítunk; készségeket alakítunk ki! A vírusokkal kapcsolatban ezerszer elmondhatjuk, hogy a gép veszélyeztetett, az adatok értékesek, hogy a számítógép — elnézést a profán hasonlatért — nem utcalány, nem lehet büntetlenül bármit beledugni stb. Mindenki akkor fog erről aktív tudást nyerni, amikor először formázza újra a winchesterét... Ezzel egyúttal a backup fontosságáról is élménydús meggyőződést szerez. Igaz ez az egyszerű copy parancsra is. Akkor tanulja meg tanulónk, ha szüksége lesz rá, és mikor már nem kell — mert például rászokik a Norton Commanderre —, elfelejti; csak az általános emlék, benyomás, információ marad meg: másolni lehet.)

Paradoxon?

De! A következő kijelentésem formálisan az előző pont ellentéte: szövegfeldolgozás csak szövegfeldolgozó programmal végzett munkával tanítható. Akkor viszont szükséges az adott szerkesztő használatát is megtanulni, különben hol a gyakorlat? — nem lesz képes magát a programot működtetni a tanuló.

A „nagy kérdés", hogy mit kezdjünk e szépen fejlett paradoxonnal: alkalmazás kontra gyakorlat. Nem akarhatjuk a gyerekbe beledrótozni valamelyik kedvenc programunkat, hiszen szinte biztos, hogy két-három év múlva nem azzal találkozik a munkahelyén. (A másodikon meg nem azzal, amivel az elsőn.) Egy alkalmazást azonban csak konkrét programmal dolgozva lehet megismerni.

Nincs egyetlen jó válasz. A két szélsőség, amivel találkoztam:

A) Kínlódjon meg vele, én is megkínlódtam... (Egy és csak egy konkrét programot tanítsunk meg egy alkalmazásból. Azt viszont nagyon alaposan. Ha már egyen túl van a gyerek, a második majd könnyebb lesz.)

B) Tapasztalat teszi a mestert... (Minél több konkrét alkalmazást kell megtanítanunk; alváltozat, bár ez ritka: párhuzamosan többfélét kell — ezekből a gyerek képes leszűrni a számára hasznos információkat. És: csökken a való-

színűsége, hogy ismeretlen programmal találkozik.)

Az én véleményem, ami szerint immáron két éve próbáljuk „megvalósítani magunkat” és elérni céljainkat, a következő: az alkalmazás lényegét, azokat a tulajdonságokat, amelyeket minden idevonatkozó programnak bírnia kell, egy egyszerű, modellnek tekinthető programon meg kell tanítanunk. Ha sikerül olyan programot találnunk, amelyik működése közben nem „fedi el” önmaga szabályaival az adott alkalmazás szabályait, az kiváló. (Nekünk még nem mindenhez sikerült ilyet lelnünk.) Ha megtanítottuk a modell használatát, adjunk sok feladatot, hogy tapasztalatra tegyenek szert. Ezután fel kell őket szólítani, hogy a rendelkezésre álló programválasztékból gusztusuknak megfelelően válasszanak (ha van időnk, többet is egymás után), és azt ismerjék ki. Ehhez persze ismét feladatokra van szükség.

A módszer hátránya egyben az előnye. Feltételezi, hogy minden programról készült egy olyan anyag, amelyikből egyedül is elsajátítható a kezelése. Megszűnnek a frontális órák, mindenki maga dolgozik, a tanár pedig valóban egyénileg foglalkozik tanítványaival. Megtanul a gyerek tanulni, egy ismeretlen programot felderíteni, használatba venni. Problémát old meg!

Az arany középúton

A módszer előnye persze a hátránya is. Olyan feladatokat kell készíteni, amelyek nem program-, hanem alkalmazás-centrikusak. A számonkérésnek is végeredmény-központúnak kell lennie. A tanárnak sok konkrét programot kell készségszinten ismernie. Nem lehet „egy gép, két gyerek” sémával dolgozni, de nagy létszámmal sem. Tapasztalataim szerint az ilyen tanulócsoport minimuma 6 fő. Optimális a 8-9, a felső határ pedig — a tanártól függően — 12-15 gyerek. Nos: tanítsunk vagy ne tanítsunk programozást? Visszakérdezek: miért ne, ha ráérünk? Miután becsületesen felkészítettük tanulóinkat az alkalmazásokból... De soha ne feledjük: programozást oktatva mi egy szakma (szakmánk?) műkedvelőit vagy — rossz esetben — kontárait képezzük. A fő cél a logikai képességek fejlesztése, a „mi történik, ha...” típusú kérdésekre a válaszok megkeresése. A logika természetesen nem az egyetlen gondolkodási mód, nem válaszol az embert foglalkoztató legalapvetőbb kérdésekre, a „mi dolgunk a világban?” típusúakra. De számomra a megfordítása szintén igaz a kijelentésnek: azzal nem is érdemes foglalkozni, amit a logika elemzési módszereivel nem vizsgáltunk meg — vagy másként: ha valamilyen rendszerben logikailag hiányosak az alapfeltevések, az a rendszer nem lehet igaz. Azonban sose veszítsük szem elől, hogy az ember lelkét igazán izgató kérdések logikailag tiszta paradoxonhoz vezetnek — akár erről is felismerhető, hogy jó felé tapogatódzik gondolkodásunk.

Albu László

Attitűdök és viszolygások

Ez a kis iromány az asztal túlsó oldalán íródott: egy tinédzser gondolkodott el tanulótársai és tanuló kortársai, valamint az őket okító tanerők viszonylatairól, továbbá a viszonyíthatóságról...

Egyáltalán hol kezdődik? — mármint a beállítódás. Az öcsém már az általános iskolában tanul számítástechnikát. Hogy ezen belül mit? Egyből bele a közepébe: programoznak, igaz, egyszerű nyelven, a Basicben, de lehet, hogy ez „kényszerűségből” fakad — ugyanis a C+4 ROM-jában megtalálható egy Basic. Van-e értelme? Talán annyi, hogy egy kis programozásfelépítést (-struktúrát?) tanulnak, bár a C+4 Basic rendszere nem éppen szerencsés, de ez van, erre van lehetőség. Mi lesz egy picit komolyabb feladat megoldásánál? Valószínűleg csodálkozás, majd összekavart, áttekinthetetlen program. S később? — mi lesz, mi lehet a „kész csoda, hogy..."?

A mi iskolánkban (de gyanítom: máshol is) háromféle embertípus létezik:

— Az első, akit abszolút nem érdekel a számítástechnika. (Pedig nálunk több tantárgyhoz is kapcsolódik a számítógép.) Ha bármely program hibázik, akkor ők már toporzékolnak, szidják az összes programozó valamennyi felmenőági rokonát. (A dologhoz hozzátartozik, hogy sok a saját fejlesztésű programunk, s némelyik még gyakran hibázik.) Ezek az emberek könnyen feladják.

— A második, akinek már nem kell betölteni a programokat, bár ezt a problémát már megoldották a batch fájlokkal. Ők azért valahol kíváncsiak rá, hogy mitől indul, amit használnak. Könnyen „felfűthetőek” egyszerű problémák megoldására, később esetleg még bele is szeretnek a számítógépbe, mint egykoron jómagam.

— A harmadik, ahová én is tartozom. Mi talán már értünk a géphez. Ahányan vagyunk, szinte annyifélét tanulunk. Közöttünk felmerült a kérdés: van-e ráció abban, hogy igazándiból megtanuljunk programozni? Hiszen sorra jönnek ki a negyedik generációs programok, s noha ezeknél a program széppé tétele, extrákkal való megtűzdelése általában megoldhatatlan, de — és ez a mai világban igen fontos — rengeteg időt takaríthatunk meg velük.

Mégis! — olyan izgalmas dolog, amikor az ember keze alól sok-sok munka után végre valami érdekes jön ki. Bár megjegyzem, addig a legjobb, amíg az ember írja. Utána csak „van”, míg nem ír egy „szuperprogramot”, amelyet azután más is sokat futtat. (Ilyet nekem még nem sikerült, ámbátor olyan program már találtatott, amit én felhasználtam...)

A tanárnak a legnehezebb dolga az első csoporttal van. Ha egyáltalán megjelennek az órán (nálunk órára járni nem kötelező, de kell), akkor is nehéz „szántásra” fogni őket, nincs igazán, ami lelkesítené ezt a társaságot. A második csoport ideális: ők dolgoznak s — ha kész vannak — mennek.

De miért nem a harmadik csoport a legkönnyebben kezelhető, kérdezhetné bárki. Nos, őszintén: hát mi sosem a feladatot csináljuk... S mi a „legszebb” a tanár munkájában? Talán: hogy leledzenek olyan egyének is „Alternatív Közgazdasági Gimnázium”-unkban, akik az ötödfeledik vagy a harmadfeledik alcsoportba lennének sorolhatóak...

— Csocsó —

Felső szintre törekvő felsőoktatás

Információk az informatikusképzés infrastruktúrájáról

A felsőoktatásban a számítástechnika, a számítógép szerepe jóval összetettebb, mint a közoktatásban és a középfokú szakképzésben. Az egyetemeken és a főiskolákon általános értelmiségiképzés, szakemberképzés, át- és továbbképzés, az oktatáshoz kapcsolódó tudományos kutatás folyik, s az intézmények irányítási, vezetési, szervezési és gazdasági feladatai is lényegesen nagyobbak és bonyolultabbak. Az oktatásban — amelynek tartalmát maguk az intézmények határozzák meg — a számítástechnikai követelmények a képzési céltól függenek.

Az általános értelmiségiképzés részeként minden egyetemi és főiskolai hallgatónak meg kellene tanulnia használni a számítógépet, képesnek kellene lennie adatbázisokat lekérdezni, szöveget szerkeszteni, elektronikus levelezést folytatni. Az értelmiségi foglalkozások szinte mindegyikének velejárója lesz maholnap a számítógép rendszeres használata — a mérnök, a közgazdász, a matematikus, a fizikus, a geológus, a térképész, a szociológus stb. munkájának már ma is nélkülözhetetlen eszköze a számítógép! Ennek a képesítési követelményekben is ki kell fejeződnie, aminek természetesen az a feltétele, hogy a képzéshez legyenek meg a szükséges eszközök: számítógépek, nyomtatók, lapolvasók, szoftverek stb.

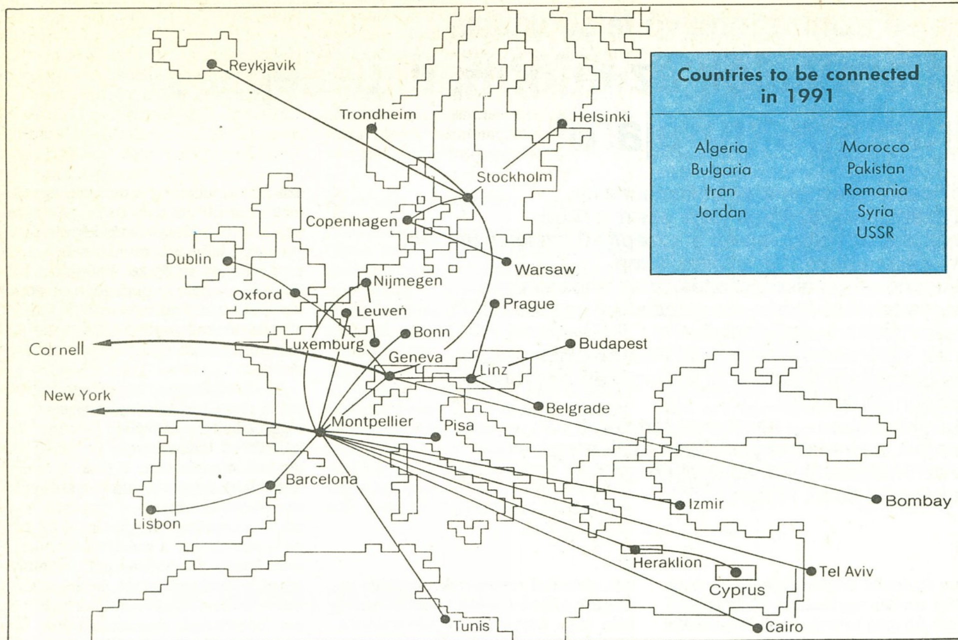
Az egyetemek és főiskolák jelentős részében olyan szakemberek képzése (is) folyik, akiket gyűjtőnéven informatikai szakembereknek nevezünk: például programozó matematikusok, programtervező matematikusok, rendszerszervezők, mérnökinformatikusok, számítástechnika szakos tanárok, hogy csak a legfontosabb kategóriákat említsük. Az ő kiképzésük eszközigénye nyilván jóval nagyobb annál, mint ami az általános értelmiségiképzéshez szükséges: a hallgatók számához viszonyítva több személyi számítógép, munkaállomások (workstationok), nagyszámítógépek, számítógépes hálózatok, különféle felhasználói és rendszerszoftverek, szakértői rendszerek, speciális perifériák (például mérő-érzékelő eszközök, nagy teljesítményű háttértárolók, rajzgépek).

„taní-tani"

Kiemelten kell szólnunk a pedagógusképzést folytató intézményekről: a tanárképző, a tanítóképző, az óvónőképző főiskolákról, a tudományegyetemek tanárképző karairól, illetőleg tanári szakjairól, valamint a műszaki egyetemeken és főiskolákon folyó mérnöktanár-, illetve műszakitanár-képzésről. A pedagógusképzésben részesülő 37,5 ezer hallgató az összes hallgató 1/3-át teszi ki. A holnap pedagógusai döntő tényezői a számítástechnikai kultúra elterjesztésének, a jövő számítástechnikai/informatikai szakemberei felkészítésének.

A felsőoktatás egészében minél előbb el kell jutni oda, hogy a számítógép alkalmazása váljék általánossá a különböző tudományok, illetőleg tantárgyak oktatásában. Ennek személyi feltételei közé sorolhatjuk a szemléletet. Az intézmények — ma már legitim — vezetőinek meghatározó többsége tisztában van a számítástechnika, a számítógép alkalmazásának jelentőségével, szükségességével, és elismerésre méltó erőfeszítéseket tesz az intézményi lehetőségek megteremtéséért, fejlesztéséért. E vezetők ilyen törekvéseikben leginkább az intézményi számítástechnikai szakemberek és a számítógépet ismerő, azt alkalmazó vagy alkalmazni akaró oktatók segítségére számíthatnak. Az oktatók között azonban még nem kevesen vannak olyanok, akik — elsősorban azért, mert nem vagy nem eléggé ismerik — idegenkednek a szá-

A főiskolák és az egyetemek számára az EGP felsőoktatási részprogramjának két utolsó éve (1989 és 1990) hozott jelentős előrelépést. A közel 500 millió Ft állami támogatás — és természetesen a saját erőfeszítések eredményeként — 1990 végén már kereken 6 ezer személyi számítógép volt az intézményekben. Ha ezt a számot összevetjük a felsőoktatásban dolgozó oktatók és kutatók 16 ezer fős, továbbá a hallgatók 102 ezer főt meghaladó létszámával, akkor nyilván nem lehetünk elégedettek, de hiba lenne az is, ha lebecsülnénk az elért eredményt. Végül is ez a hatezer számítógép képezi a továbbfejlődés alapját.
Magyarországon az 1986-ban indult IIFP (Információs Infrastruktúra Fejlesztési Program) keretében az utóbbi években közel 160 kutatóintézet, felsőoktatási intézmény, könyvtár és múzeum kapcsolódhatott be a postai vonalakon létesült adathálózatra telepített IIF rendszer szolgáltatásaiba, amelyek közül a legfontosabbak: a hazai és a nemzetközi elektronikus levelezés, valamint a hazai adatbázis-szolgáltatás. Az információátvitel sebessége 2400 vagy 9600 bit/s. A rendszer fő erőforrása, egy IBM 4381-es számítógép a Magyar Tudományos Akadémia (MTA) Számítástechnikai és Automatizálási Kutatóintézetében (SZTAKI) üzemel, s ugyanitt működik az az IBM 4341-es gép, amely az IIF-hálózatot összekapcsolja a linzi EARN (European Academic Research Network) csomóponttal.



mítógéptől. A tapasztalatok alapján arra lehet számítani, hogy a tárgyi feltételek javulásával ez egyre kevesbé lesz így.

Cél és eszköz

Az oktatáshoz kapcsolódó tudományos kutatási tevékenységhez, továbbá az intézmények tanulmányi, gazdasági és egyéb vonatkozású irányítási feladataihoz, valamint a nemzetközi kapcsolatokkal összefüggő teendők ellátásához szükséges számítástechnikai eszközigényre nem térünk ki. Csupán annyit jegyzünk meg, hogy egyes — például a fizikai — kutatásokhoz rendkívül gyors, nagykapacitású operatív és háttérmemóriával rendelkező számítógépek is kellenek (egyebek mellett), a nemzetközi oktatási és kutatási együttműködéshez pedig elengedhetetlen a külföldi egyetemek, kutatóhelyek, adatbázisok számítógépes hálózatokon keresztüli elérése.

Budapesten — a Művelődési és Közoktatási Minisztérium 34 millió Ft-os támogatásával — már épül az az optikai kábeles gerincvezeték, amely összekapcsolja a Budapesti Közgazdaságtudományi Egyetemet /BKE/, a Budapesti Műszaki Egyetemet /BME/ és az Eötvös Loránd Tudományegyetemet /ELTE/. Folynak az előkészületei a Semmelweis Orvostudományi Egyetem /SOTE/ és az ELTE hálózati összekapcsolásának, s napirendre került a Kertészeti és Élelmiszeripari Egyetemet /KÉE/ a BME-vel összekötő optikai kábel kiépítése is. Az építkezést tovább folytatva fokozatosan létrejöhet egy budapesti, tehát nagyvárosi felsőoktatási számítógépes gerinchálózat.

Ez a hálózat elősegíti azt, hogy a fővárosi egyetemek egy része elérhesse azt a — Kelet-Európában jelenleg a legnagyobb teljesítményű — IBM 3090-170J típusú számítógépet, amelyet az IBM 1991.június 19-én hozott be Magyarországra, a BKE Számítóközpontjába telepített, és 1991.szeptember 27-én ünnepélyesen átadott használatra a BKE-nek, a BME-nek és az ELTE-nek.

1990-ben — az EGP felsőoktatási részprogramjának keretében — prioritást kapott a lokális számítógép-hálózatok építése. A Művelődési Minisztérium kezdeményezésére és anyagi támogatásával Pécsett, Debrecenben és Szegeden városi felsőoktatási hálózat tervezési munkálatai kezdődtek meg. E hálózatok létesítését az indokolja elsősorban, hogy a felsorolt városokban az egyes egyetemek, főiskolák épületeinek földrajzi szétszórtsága miatt az intézményi hálózatok átfednék egymást, s gazdaságosabb lehet olyan városi gerinchálózat kiépítése, amely valójában az intézményi belső hálózati és az intézmények közötti információátviteli funkciókat is elláthatja.

Pillanatok alatt, iszonyatos távolságokon át

A fejlett országokban ma már a személyi számítógépek százezrei működnek lokális és magasabb szintű hálózatokba kapcsolva. Az egyetemek oktatói, ku-

tatói, hallgatói e hálózatokon keresztül elektronikus levelezést folytathatnak, közeli és távoli, hazai és külföldi adatbázisokat (könyvtárakat, szabadalmi tárakat, névtárakat stb.) érhetnek el, nagy teljesítményű számítógépekhez férhetnek hozzá, helyenként több millió bit/s sebességgel.

Nyugat-Európában, Észak-Amerikában, Japánban, Ausztráliában nemzeti egyetemi számítógépes hálózatok vannak, s ezek egymással is össze vannak kapcsolva. Az európai egyetemi hálózatokat 1984 óta az EARN (European Academic Research Network) fogja össze — lásd az ábrát —, amely Franciaországban, a montpellier-i egyetemen kapcsolódik az USA-beli BITNET-hez, a kanadai NORTHNET-hez és az ázsiai ASIANET-hez. Ennek a nagy interkontinentális rendszernek 1990 végén 45 országban 3055 csomópontja volt. Magyar vonatkozásban Linzet kell példának kiemelnünk, mivel 1990 októberében itt jött létre kapcsolódási lehetőség számunkra.

A közgyűjtemények is

Az IIFP 1991. június végén zárult első szakaszában 21 egyetem, 8 főiskolai kar és 8 főiskola lett IIF-tagintézmény. Az 1991. július 1-jétől 1994-ig tartó második szakaszban elvileg az összes többi felsőoktatási intézmény s a könyvtárak, múzeumok, levéltárak (ezeket együtt közgyűjteményeknek nevezzük) egy további része is csatlakozhat az épülő országos rendszerhez.

A lokális hálózatok létesítésében már eddig is több felsőoktatási intézmény ért el nagyon szép eredményeket. Ezek teljes körű felsorolása helyett csak néhány kiemelkedő példát említünk meg. A Budapesti Műszaki Egyetemen a Villamosmérnöki és a Gépészmérnöki Karon már kari hálózatok működnek, s ezek rajta vannak az egyetemi gerinchálózaton. A szegedi József Attila Tudományegyetemnek eddig három olyan épülete van optikai kábelekkel összekötve, amelyekben tanszéki, számítóközponti, hivatali és könyvtári hálózatok üzemelnek. A nyíregyházi Bessenyei György Tanárképző Főiskolán teljes intézményi hálózat működik, amely átfogja az oktatási létesítményeket, a tanulmányi és a gazdasági hivatalt, a könyvtárat, valamint a kollégiumot. A három épületet itt is optikai kábelek kötik össze.

A magyar egyetemek, főiskolák és kulcsfontosságú kutatási bázisaik, a közgyűjtemények információs infrastruktúrája azonban csak akkor lesz képes a fejlett országokban ma már általános szolgáltatások nyújtására, ha az intézményeken belül kiépülnek azok a lokális számítógépes hálózatok, amelyek

— megfelelő számú, teljesítményű, kiépítettségű személyi számítógépet, munkaállomást, nagyszámítógépet és digitális telefonközpontot fognak össze;

— nagy sebességű digitális adat-, hang- és képátvitelre potenciálisan egyaránt alkalmasak;

— a nemzetközi szabványoknak megfelelő kialakításuk, kompatibilitásuk, egyeztetett szoftvereik révén országos hálózatba illeszthetők.

Az intézményi hálózatok különböző szintjei a következők: 1. végfelhasználói, 2. tanszéki (klinikai, intézeti), 3. kari, 4. egyetemi, főiskolai hálózat. E hálózatoknak célszerű átfogniuk az intézményi — esetleg a kari — könyvtárakat, a tanulmányi és a gazdasági hivatal(oka)t és a kollégiumo(ka)t, a tanárképzést folytató intézmények esetében pedig a gyakorló iskolákat is.

A felsőoktatás számítástechnikai feltételrendszere a következő években várhatóan nagymértékben fejlődik. Az egyetemek és főiskolák ehhez — remélhetőleg — jelentős támogatást kapnak az Információs Infrastruktúra Fejlesztési Program keretében, a Felzárkózás az Európai Felsőoktatáshoz Alapból, a Központi Műszaki Fejlesztési Alapból, más alapokból és alapítványokból, valamint a számítástechnikai cégektől. (Ez utóbbira egy egészen friss példa: a DEC hét egyetemnek és főiskolának ajánlott fel nagy teljesítményű számítógépeket rendkívül nagy árkedvezménnyel.)

Engloner Gyula

A hazai programozóképzés fellegvára(i)

A tudás nem pottyan az égből!

Magyarország egyetemi városai közül 3 helyen: Budapesten, Szegeden és Debrecenben folyik profi programozómatematikus-képzés. Cikkünk azoknak szól, akik az oktatás indulásáról, irányairól, erősségeiről és a tervekről szeretnének tájékozódni.

A 70-es évek elején a vidéki egyetemekkel együtt és velük egyeztetett tematika alapján megindult az ELTE Természettudományi Karán is a számítástechnikai szakemberek képzése. Mivel Magyarországon a számítástechnika matematikusközegben fejlődött ki, így az oktatásra is ez a szemlélet nyomta rá a bélyegét. Ezt tükrözte a programozó matematikus szak elnevezés is; ma már azonban informatikusképzésről hallhatunk.

A centrális „erődítmény"

Az informatika szakon a hallgatók olyan problémák számítógépes megoldására készülnek fel, amelyekhez mély matematikai ismeretekre van szükség. A képzés mindazonáltal az élethez szabott: képesség és készség szerint variábilis. A hallgatók az első lépcső után főiskolai programozó matematikusi diplomával, végzett szakemberként akár el is hagyhatják az egyetemet. Az ambiciózusabb vagy a „jobban eleresztett" fiatalok azonban folytathatják tanulmányaikat a második lépcsőn, a programtervező matematikus szakon azzal a céllal, hogy magasabb szintű kutató, fejlesztő vagy irányító feladatkört is elláthassanak. Lehetőség van továbbá posztgraduális képzésre is, amely a specializálódást alapozza meg egy-egy aktuális témakörben. Az erre vállalkozók munka mellett tanulnak, és a szakdiplomával együtt doktori címet is kapnak.

Természetesen külföldieket is oktatnak az egyetemen. Ők az első lépcsőben B.Sc (Bachelor of Science) diplomát, míg a második lépcsőben M.Sc (Master of Science) szakdiplomát kapnak. Az egyetemen folyik tanárképzés is. A hallgatók maguk dönthetik el, hogy egymagában vagy párban veszik-e fel a számítástechnikai tanár szakot.

Az informatikai szak 2. lépcsőjében egy olyan sávrendszert alakítottak ki, ahol a tantárgyakat láncban lehet felvenni. Egy minimális, mindenki számára kötelező magon kívül 6 sávból, témakörből legalább négyet kell választaniuk a hallgatóknak. A szakosodás stúdiumain kívül speciális kollégiumok keretében olyan témákkal foglalkoznak, amelyek még nem kristályosodtak ki. Így az egyes szakterületek a speciális kollégiumok tapasztalatain keresztül kerülnek be az egyetemi általános oktatásba.

Kellemes meglepetés, hogy az egyetem oktatói milyen liberális eszméket képviselnek és érvényesítenek az oktatásban. Azt is idejekorán ismerték fel, hogy a képzést rugalmasabbá és tagoltabbá kell tenni, hisz a diákok életközege mindig változik. Például ha a környezet sok jó, magasan kvalifikált szakembert igényel, akkor ez visszahat az egyetemi képzés időtartamának megválasztására is — úgy, hogy a munkába állást nem halogat(hat)ó diákok majd tovább tanulhatnak akár a második lépcsőben, akár a posztgraduális képzés keretében. Az egyetem oktatói rájöttek arra is, hogy a hallgatók egyetemi teljesítményét nem a teljesített félévek számában, hanem az elsajátított tananyag mennyiségében kell mérni.

A magas szintű kvalifikációt segíti az is, hogy a megfelelő nyelvtudású diákok a harmadik, negyedik vagy az ötödik évben féléves kurzuson vehetnek részt Nyugat-Európa valamelyik egyetemén vagy kutatóintézetében. A múlt tanévben a közel 40 főnyi negyedévesből 14-en tanultak külföldön, az idén pedig ez a szám már a 30-at is elérheti.

A tudás átadását és elmélyítését a fejlett eszközbázis is segíti. Az egyetemi számítóközpont mellett a diákok MicroVAX és Schneider gépeken dolgozhatnak. Természetesen hálózat is működik az egyetemen, sőt a munkaállomások is kiépülőben vannak. A szövegszerkesztők és a lézerprinterek változatainak megismerése által a diákok már az egyetemen megtanulják azt is, hogy munkájuk eredményét be is kell csomagolniuk, vagyis a tartalom, az ötlet rangján túl a megoldás külalakjára is kell adniuk.

Az ELTE-nek mint műhelynek a rangját bizonyára az is emeli, hogy a hallgatók közül többen — Kelet-Európából elsőként — már nemzetközi programozói versenyeken is részt vettek, ahol jó helyezéseket értek el. Emellett a külföldi részképzésen megszerzett elismerő minősítések is azt bizonyítják, hogy jó alapokkal jöttek a magyar hallgatók. A diákok mellett azonban az informatika tanszék munkatársai is öregbítik a TTK hírnevét: sikeres oktató-kutató munkát végeznek külföldön. A 40 tanszéki dolgozó közül öten-hatan rendszeresen dolgoznak hazánkon kívüli egyetemeken és kutatóintézetekben.

Az „alaprajzok" eltérései

Ha a három egyetemen folyó oktatómunkát összevetjük, akkor úgy tűnik, hogy a leginkább gyakorlati jellegű képzés a budapesti egyetemen folyik. Az ELTE mögött olyan kutatóintézetek (SZTAKI, KFKI, SZKI) állnak, amelyek bizonyos alkalmazási területet képviselnek. A szegedi egyetemre egy-egy meghatározott irányba mutató elméleti képzés a jellemző. A debreceni egyetem pedig az elméleti és a gyakorlati képzés kiegyensúlyozásában vezet — ha mindenáron viszonyítani akarunk. Az biztos, hogy egyik fellegvárból sem lehet a diplomákat a légből elkapkodni...

A programozó matematikus szakra évről évre egyre többen jelentkeznek. A túlkínálatnak köszönhetően az első évre felvett hallgatók több mint 10%-a maximális pontszámmal került be az egyetemre. A szak népszerűségét, társadalmi elismertségét jelzi az is, hogy jelenleg a TTK létszámának kb. 20-25%-a informatika szakos hallgató.

Sziebig Andrea

A Basic nyilaitól ments meg Uram minket!

Néhány év újgyarmatosításért...?

Áldás vagy átok a személyi számítógépek elterjedése az oktatásban — bizonyára sokáig vitatott kérdés marad szakmai és szülői körökben. Abban azonban aligha kételkedik bárki is, hogy legalább egy szakterületre, a programozó- (szoftveres?, informatikus?) képzésre csak pozitívan hatott. Vagy mégsem? Zajki László, az SZKI egykori osztályvezetője, néhány éve egy határral nyugatabbra fejleszt szoftvereket. Van véleménye a dologról — kollégánk pedig lejegyezte az elmondottakat.

Mindenképpen érdemes megvizsgálni ezeket a hatásokat, hiszen ma és a közeljövőben ez a PC-s generáció határozza meg a szakma jövőjét. Komoly bajban lennénk azonban, ha mérleget kellene vonnunk. Egyfelől számtalan pozitívumot fel lehet sorolni, másrészt azonban a kisgépek elterjedése erősen visszavetette a programozás — nem igazán jó szó: — fejlődését.

S noha úgy tűnik, két külön feladatkörről van szó, ha úgy tetszik, egy szakma két ágáról, ez csalóka szempont. Addig, amíg játszunk a PC-kkel, nincs semmi gond. Ám ha megnézzük, mi mindenre alkalmazzák ma ezeket a — már a nevükkel is figyelmeztető: személyi — számítógépeket, kiderül: a szerepkörnek csak a töredéke játék, s a tipikusan egyfelhasználós feladat is csak a kisebb rész. Nehéz problémák ugyanazzal a logikával, programozói filozófiával vannak megoldva, amelyen kívül — s itt jön az oktatás szerepe — mást nemigen tapasztaltunk (legalábbis a közelmúltig) az egyetemen. Itt egyébként nemcsak a barkácsolmányokról van szó, nagy cégek piacvezető termékeiről is.

Az oktatásnak szerintem az ismereteken felül szemléletmódot, stílust és — merem használni: — erkölcsöt is tanítania kell. Ha nincs meg a hardveres védelem, az operációs rendszer biztonsága, csak a szoftveres önkorlátozása védheti meg a felhasználót a csalódásoktól. E hozzáállás hangsúlyozását hiányoltam a magam és a fiatalabbak tanulmányaiból. Az alapprobléma, persze, az operációs rendszerek és a támogatói hardverek tehetetlenségi vagy üzletpolitikai okokból lassú fejlődésének köszönhető, bár lassan-lassan, az igények, a felhasználások és természetesen a hardver lehetőségeinek bővülésével nő a késztetés a nagygépeken már bevált technikák bevezetésére (lásd XENIX, WINDOWS).

Le kell szokni az egyfelhasználós gépeken elkövethető akciókról. A mai rendszerekben egyetlen programozó sem látja át teljesen a folyamatokat. A megbízható működést csak úgy lehet garantálni, ha senki sem lép ki a saját programszintjéről, nem próbál megoldani más feladatot, mint amit kell, azt viszont — közérthetően dokumentálva — tökéletesen. Ez a nagygépes világban mindig is egyértelmű volt, ám ezt a kultúrát a PC-ken folytatott bitvadászat, illetve a megfelelő eszközök, például a precíz fordítók hiánya újra megtanulandóvá tette.

A PC-korszak elején maguk a programozási nyelvek tették lehetetlenné a jó programok írását: a PC-k előtt kialakuló ALGOL irányzatot elsöpörte a minik assembler- és Basic-inváziója, amihez képest a Pascal és a C színrelépése óriási ugrás volt. Ám a további előrelépés — például a C++ felé — már rettenetesen nehezen megy, az eddig felhalmozott tudás konzervatívvá teszi a cégeket.

Azt gondolom, a magányos farkasok — Szilícium-völgyet alapító, négy-öt éve még virágzó — kora véget ért. Nyugaton az alapszoftver (ahonnan még látszik a gép „belseje") írásával foglalkozók száma is lecsökkent. A jövő mindenképp a teamekben dolgozó —azt mondanám: — informatikusoké. Precízebben: kialakul az a gyakorlat, hogy egy szakértői csoport definiálja a feladatot, megfogalmazza az igényeket és lehetőségeket. Egy másik analizátorcsoport szétnéz, hogy mit lehet kapni a szükséges programokból (s inkább vesznek öt, az igényeket lefedő készet, mintsem integráljanak és kiegészítsenek hármat). A fennmaradó feladatokra pedig vagy egy azonnal kódolható leírást készítenek, vagy egy specifikációt, amelynek alapján elkészíttetik a programot.

Ezért egyre jelentősebb a dokumentáció szerepe — és ezt is meg kell tanítani, mármint a dokumentálás korszerű módozatait. Egyáltalán: azt a lazaságot kellene megszüntetni, hogy a szoftver megírásával véget ér a programozó munkája. A termék élete akkor kezdődik csak igazán, amikor kész a dokumentációja is — garancia, továbbfejlesztés stb.

Az ember nyugati útjai során általában tapasztalja: a cégek a hazainál jóval kevesebb diplomás mérnököt, programozót foglalkoztatnak, több viszont a technikusi és az alacsonyabb végzettségűek, valamint a menedzserek száma. Én egy szoftverházban dolgozom, ahol az arányok nem ilyenek. Tény viszont, hogy nem csak Magyarországon fájó a „túlképzettség". Sok a favágómunka: ha a megrendelő fizet hatmillió sor átírásáért COBOL 2.0-ból 2.3-ba, akkor azt meg kell csinálni. Az is igaz, hogy a kinti egyetemek kevésbé mély és széles áttekintést nyújtanak, viszont a szakirányú gyakorlati oktatás szinte azonnal bevethetővé teszi a végzetteket. A piaci viszonyok a mi oktatásunkat is erre — sőt: mivel a világcégek hozzák magukkal a szoftverfejlesztéseiket is, még inkább az alkalmazói kiképzés felé — fogják eltolni.

Kérdés azonban, hogy fel szabad-e adni néhány(?) év újgyarmatosításért az elméleti szakemberek képzését.

Aje

Műszaki informatika

Reál(is) értelmiségiek az évezred végén

A műszaki informatika a hazai felsőoktatásban alapított új mérnöki szak, amely az informatika és valamely műszaki terület (villamosmérnöki, gépészmérnöki stb.) érintkezési felületét fedi le határterületi szakként. A főiskolai végzettségűek okleveles üzemmérnök-informatikus, míg az egyetemi tanulmányaikat eredményesen befejező hallgatók okleveles mérnök-informatikus végzettséget szereznek.

A szak egyike a műszaki felsőoktatásban 1989-ben definiált alapszakoknak, melynek hazai gyökerei az 1985-ben a Kandó Kálmán Villamosipari Műszaki Főiskolán indított posztgraduális, illetve az 1986-ban a BME Villamosmérnöki Karán alapított informatika szakok voltak.

Az informatika társadalmi súlyának növekedését figyelembe véve a Művelődési és Közoktatási Minisztérium Műszaki Informatikai Szakbizottságot hozott létre, mely 1990 őszén fogalmazta meg a műszaki informatika szak oktatásánál érvényesíteni javasolt célokat és elvárásokat. Ezek szerint: kellő gyakorlat megszerzése után legyen képes a végzett üzemmérnök-informatikus a műszaki informatikai rendszerek előkészítési, üzemeltetési és irányítási feladatainak ellátására, a műszaki informatika módszereinek ismeretét igénylő műszaki alkotások tervezésére, fejlesztésére, létrehozására — míg a mérnök-informatikus a műszaki informatika módszereinek széles körű ismeretét és mérlegelését igénylő, komplex műszaki alkotások tervezésére, fejlesztésére, létrehozására, továbbá műszaki-gazdasági tevékenység irányítására, a szakterület tudományos kutatásaiba való bekapcsolódásra.

A szakbizottság körvonalazta az elvárt szakmai törzsanyagot is (lásd a mágneslemezen), melynek oktatására a teljes képzési idő mintegy 50-60%-át javasolta előirányozni.

A bizottság az általános elvárások között szerepelteti a tárgyi és személyi feltételeket is. A tárgyi feltételek egyik eleme az, hogy a hallgatók számára a teljes képzési idő alatt heti legalább 8-10 óra géphasználatot kell biztosítani. Munkahelyeken jelenleg PC/AT kategóriájú személyi számítógépek értendők, melyek minimális számára a hallgatói összlétszám/5 elvárás fogalmazható meg. Megfelelő ellátottsághoz ennek kb. kétszerese szükséges. A PC-ket célszerű helyi hálózatban üzemeltetni. Szükséges továbbá, hogy kellő számú nyomtató, rajzoló, grafikus terminál stb. álljon rendelkezésre.

Az oktatáshoz PC-kategória feletti eszközök (mérnöki munkahelyek, VAX- vagy IBM-mainframe) is kellenek. A tárgyi feltételek fontos eleme a legálisan beszerzett szoftverkészlet — kellő számú installáción vagy a hálózaton való futtatás jogával. A műszaki informatika elfogadható színvonalú oktatásához a hallgatók számára az órarenden belüli géphasználaton túl szabad gépidőt is biztosítani kell, valamint legalább alkalmanként a nemzetközi adatbázisokhoz való hozzáférést.

Tekintettel arra, hogy a műszaki informatika határterületi szak, és a végzett hallgatók — a tudomány egy interdiszciplináris területén — mérnökök lesznek, számítógépek mellett a klasszikus mérnöki munkára felkészítő további laboratóriumokat is biztosítani kell, továbbá egy aktuális állományú szakkönyvtárat is.

A személyi feltételek tekintetében az egyetemi szintű oktatás előfeltétele a tudományos iskolát teremtő és folyamatosan gondozó egyéniségek, azaz a „professzorok" jelenléte és aktív közreműködésük egyes tárgyak előadásában, valamint az egész szak gondozásában. A nemzetközi gyakorlatnak megfelelően célszerű előírni, hogy legalább négy iskolateremtő egyéniség vállaljon informatika-specifikus tárgyban tematikakidolgozást és előadást.

Hasonló minőségi elvárásokat kell a főiskolákon is megfogalmazni. Hosszú távon a főiskola rangját a vezető oktatók színvonalas tudományos-szakmai tevékenységének kell megalapoznia; legalább három elismert tudományos-szakmai egyéniség aktív részvétele a műszaki informatika tárgyköreinek oktatásában elengedhetetlen. A gyakorlati foglalkozásokhoz és a tervezési feladatokhoz ennél lényegesen nagyobb létszám — csupán a diszciplínák sokféleségéből adódóan is minimálisan 10-15 fő — szükséges. Mégsem tartjuk célszerűnek a teljes létszám és a felkészültség általános előírását, mert az sok tényező, többek között az oktatás finanszírozási módszerének is függvénye.

A szakbizottság a szak alapítását a lemezmellékleten felsorolt felsőoktatási intézményekben javasolta — az ugyancsak ott feltüntetett létszámokkal. Hogy a műszaki informatika szak oktatásáról konkrét képet is adjunk, a lemezen két implementációt ismertetünk: a BME Villamosmérnöki Kar egyetemi szintű, illetve a KKMF főiskolai szintű műszaki informatika tanterveit. A hallgatóknak tanulmányaik során két szakterületi modult kell felvenniük. A modulok kínálata az igényekhez alkalmazkodik, és nemcsak a Villamosmérnöki Kar, hanem az egyetem más karai is meghirdetnek modulokat. Megadjuk néhány modul felépítését is.

Sem az egyetemi, sem a főiskolai tanterv nem rögzíti a szakterületi modulok és a fakultatívan felvehető tárgyak választékát — a mindenkori igények és lehetőségek szabad teret kaptak. A két tanterv közötti átjárhatóság feltételeit a két intézmény most egyezteti, hogy a főiskolát végzett hallgatók jó tanulmányi előmenetelük esetén továbbtanulhassanak a BME Villamosmérnöki Karán. Az egyetemet kiváló szinten teljesítők pedig a posztgraduális doktoranduszképzésben vehetnek majd részt.

Selényi Endre — Sima Dezső

Csak 18 éven fölülieknek!

Kiket és mire tanítsunk?

A 18 év alatti korosztályhoz a Számalk nem rendelkezik didaktikai tapasztalattal, tananyaggal — de ez az iskola túl drága hely is lenne nekik. Előfeltétel az érettségi. A hallgatók között viszonylag nagy az egy diplomával már rendelkezők aránya. Őket igyekeznek tudásukhoz mérten, külön csoportokban képezni tovább.

Felső korhatár nincs, az esti, levelező jellegű informatikai szakemberképzésben mégis döntően a 30-35 év alattiak vesznek részt. A rövidebb, általában egyhetes továbbképző tanfolyamokon a jelentkezők korhatára már elmosódó. A továbbképző tanfolyamok relatíve drágábbak, mint a szakemberképzők, ezért jobban mérlegelendő az előképzettség.

Szelekció — árakkal

A szakemberképzésre jelentkezőknek minden évben a nyár végén tartanak ún. előkészítőket; felvételi vizsga nincsen: az első félév eredményei döntik el a továbbjutást.

Mivel a Számalk-oktatást az állam pénzzel nem támogatja, ezt a befizetett tandíjak tartják fenn. Noha az oktatás minőségében meghatározó az oktatók személyisége, a gazdasági feltételek hiányát a legjobb előadók sem képesek pótolni. A valamely tanfolyamhoz rendelkezésre álló pénz sok mindent befolyásol: például az egy-egy hallgatóra vetítve rendszerint „olcsóbb" (nagytermes, elméleti) és a „drágább" (kistermes, számítógépes laboratóriumi gyakorlati) órák arányát — vagyis a mind önállóbb feladatmegoldások segítségével haladó korszerű ismeretátadás lehetőségét a passzív befogadással szemben.

Az Európához csatlakozás jegyében szeretnének minél színvonalasabb oktatást adni, de figyelembe kell venniük a jelentkezők (és az ő vállalataik) teher viselőképességét/fizetési hajlandóságát is: a várható kereslethez kell igazítani a megfizethető kínálatot.

A tanulást a magyar társadalom még nem tekinti belátható időn belül megtérülő beruházásnak (a megnövekedő szaktudást még mindig nehéz és bizonytalan nagyobb jövedelemre is beváltani).

A Számalk-oktatás hallgatóinak zöme a mintegy 4 és fél millió dolgozó soraiból kerül ki. Közülük már napjainkban százezrekre tehető, akik előbb-utóbb kapcsolatba kerülnek a számítástechnikával. A frissen érettségizettek számára indított, nappali — egyre inkább idegen nyelvű — képzés utánpótlását az a 70-80 ezer fiatal jelenti, aki évente elvégzi a középiskolát. Az egyetemek/főiskolák még a felvételi keretek bővülése esetén sem tudnának sokkal többet befogadni 30 ezernél — évente

Csak ambíció kérdése...

A Számalk, illetőleg oktatási jogelődje, a Számok több mint 20 éve képez számítástechnikusokat. A vállalat nemcsak Magyarország, hanem valószínűleg az egész kelet-középeurópai térség legnagyobb, iskolarendszeren kívüli oktatási intézménye: évente mintegy 8000 hallgatót fogad — 5000-et a hosszabb (szakemberképző), oklevelet adó és további 3000-et a rövidebb (továbbképző) tanfolyamain. A cég a profilját az utóbbi 3 évben menedzser- és üzletemberképző, angol, német és francia nyelvű, vállalati szintű gyakorlattal kiegészített kurzusokkal is bővítette.

A tanári kar mintegy 40 főállású és 150-200 külső oktatóból áll. Az idegen nyelveket — fakultáció keretében — rangos nyelviskolák tanárai tanítják. Budapesten kívül még számos helyen folyik oktatás: Debrecenben, Győrött, Miskolcon, Pécsett, Szegeden, Békéscsabán, Kaposvárott, Salgótarjánban, Szolnokon, Szombathelyen, Veszprémben stb. A budapesti helyszín a Szakasits Árpád úti központi épület, amely eredetileg is oktatási célokra készült: két nagy előadótermét, mintegy 15-20 PC-vel/terminállal felszerelt kis előadótermét, szakkönyvtárát, könyvesboltját, az előadótermek sorába illeszkedő számítástechnikai diákszaküzletét számos egyetem, főiskola megirigyelhetné.

A Számalk a térség első felhatalmazott Novell Oktatási Központja, és most épül ki az oktatási kapcsolat a Microsoft céggel. A holland WSG-Volmac céggel egy új, intenzív számítástechnikai képzés bevezetéséről folynak a tárgyalások. A Magyarországon már 3 éve tanító angolok, németek, franciák elhozzák hazájuk kultúráját is a szakismeretek, módszerek mellé. A menedzserképzésben partnere például a Számalknak a dán Time Manager International cég, az angol Elliot stb. Rangos külföldi előadók tartanak szemináriumokat a helyes időgazdálkodásról, az ügyfelekkel való bánásmódról, a minőségbiztosításról, a korszerű számvitelről stb. Angol (ICL) módszer alapján tartják az ún. döntési konferenciákat vállalatok felsővezetői számára.

A Számalknak 12-féle oklevél kiadására van „állami jogosítványa". Ezek:
— számítástechnikai szervező (3 különböző szakiránnyal)
— információrendszer-szervező
— számítógép-programozó
— számítógéprendszer-programozó
— információrendszer-programozó
— személyiszámítógép-kezelő
— személyi számítógépes szoftverüzemeltető
— számítógépkezelő (operátor)
— számítóközponti üzemeltetésvezető
— személyi számítógépes műszaki karbantartó
— műszaki karbantartó (számítógéprendszer)
— elektronikai karbantartó műszerész.

A több tucat személyi számítógépes, minigépes és nagygépes továbbképző tanfolyamból választók pedig tanulhatnak operációs rendszereket, programozási nyelveket, adatbázis-kezelőket, táblázatkezelőket, szövegszerkesztőket, integrált programcsomagokat, hálózatokat stb.

Az idén szeptemberben új irány nyílt: a neves Oxford Politechnickel kooperálva megkezdődött az üzletemberképzés az idegen nyelveket tudó, frissen érettségizett fiatalok számára, 4 éves nappali formában. Ezt az iskolát International Business School—Budapest-nek (rövidítve: IBS—B-nek) nevezik. Az iskola a sikeres hallgatóknak a nemzetközileg elismert, ún. BABS diplomát (Bachelor of Art in Business Studies) adja.

nagyon sok fiatal keres tehát máshol továbbtanulási lehetőséget. De a Számalk sem képes belátható időn belül évi néhány száz főnél többnek a felvételére; itt is a tandíjakkal befolyásolható fizetőképes kereslet szabja meg a jelentkezők számát.

Nagy szükség lenne már most olyan alapítványokra, kedvezményes hitelkonstrukciókra, melyek a hallgatók válláról — legalábbis átmenetileg — levennék a tanulás anyagi gondjainak egy részét. Indokolt lenne továbbá olyan adórendszer bevezetése, melyben a tanulásra fordított összegek levonhatók az adóalapból.

Válaszok és további kérdések

A „Kiket tanítunk/tanítsunk?" kérdésre már adottak a válaszok. A „Mire tanítsunk?" kérdését inkább úgy kellene feltenni, ahogyan az amerikai menedzser-tankönyvek is ajánlják: „What is our business — and what should it be?" Azaz: „Milyen üzletben is vagyunk mi érdekeltek — és tulajdonképpen milyennek kellene lennünk?"

Aki formálisan oktatást (tulajdonképpen egyfajta szolgáltatást) kínál a piacon, az valójában ajánlhat valami egészen mást is ezen a címen. A skála igen széles: lehetővé tehet, fölvillanthat például

— szabadidőt, melyet nem kell a munkahelyen eltölteni;
— sportolási alkalmakat;
— találkozási lehetőséget hasonló érdeklődésű kollégákkal;
— érdekes emberek megismerését;
— presztízst (még ezt is megengedhetem magamnak!);
— tanfolyami segédletek formájában szétosztott értékes eszközöket;
— külföldi utazási esélyt stb.

vagy ezek valamilyen kombinációját. Ha — számos országhoz hasonlóan — Magyarországon is le lehetne vonni a személyi jövedelemadó alapjából az oktatásra fordított összegeket, akár talán még bővülhetne is a kör.

A végső válasz

A Számalk a viszonylag gyorsan használható tudásra és a biztos tájékozódáshoz szükséges alapokra teszi a hangsúlyt. Nem annyira végrehajtókat, inkább döntéshozókat szeretne képezni: olyan szakembereket és olyan generalistákat, akik a megszerzett szakismeretek mellett, illetőleg ennek birtokában átlátják a társadalom egészének működését; ismerik a gazdálkodás, a vállalkozás fortélyait; jól képesek kommunikálni másokkal; képesek gyümölcsöző emberi kapcsolatokat kialakítani. A cél, hogy a hallgatók azzal a tudattal búcsúzzanak el a tanfolyamoktól: ez az a hely, ahová érdemes komplex problémák megoldásaiért fordulni. Nem azért, mert mi mindent tudunk, hanem azért, mert itt megbízhatóan megkeresik a megfelelő partnert, és az ügyfeleknek garantálják a korrekt megoldást.

This is the business we should be in — mi ebben az üzletben szeretnénk igazában benne lenni.

Kovács Ervin

Munka mellett (és után)

A Számalkban folyó, szakképzettséget/szakképesítést adó informatikai oktatás lényegében esti/levelező jellegű tanfolyamok sora — elsősorban fiatal, már munkában álló felnőttek számára. A látogatás a szorgalmi időszakban (ez évi 37 hét) általában 2 alkalomra elosztott, heti 9 órai délutáni-esti elfoglaltsággal jár. Ennek kb. fele előadások hallgatása, másik fele általában számítógépes gyakorlat. A felsőfokú oklevélért 3 év alatt hozzávetőlegesen 850 órán kell részt venniük a hallgatóknak.

A megfelelő vizsgák sikere után 12 különféle informatikai szakképzettséget igazoló oklevelet/bizonyítványt vehetnek át a hallgatók.

A tankönyveket, jegyzeteket hagyományosan a cég saját kiadója állítja elő, s ezeket helyben is meg lehet vásárolni; bárki hozzájuthat azonban e kiadványokhoz a szakkönyveket is áruló könyvesboltokban is. Bár a legtöbb tantárgyhoz jó irodalmat tudnak adni, az ismeretátadás alapvető formájának az előadásokon elhangzottak és a gyakorlatokon szerzett tapasztalatok tekinthetők. A gyakorlatok zöme számítógépes: tipikusan 2-3 hallgató ül egy PC-nél vagy terminálnál. Ez természetesen a szakemberré váláshoz a minimumot nyújtja csupán, de arra számítanak, hogy a legtöbb hallgató a munkahelyén is számítógépekkel dolgozik. Akiknek ilyen lehetőségük nincs, szombatonként külön gépidőhöz juthatnak.

Mivel a szakemberképzés tematikáját lényegében a Számalk dolgozta ki az iskolarendszeren kívüli informatikaoktatást felügyelő KSH számára, aki a Számalkban tanul, az az ismeretek forrásához jött.

Lépcsők a számítástechnikai tudáshoz

Világcégek partnereként...

Közismert, hogy az oktatás nem nyereséghozó tevékenység, mégis nagyon sok számítástechnikai vállalkozás érdemesnek tartja a képzési lehetőségek fenntartását.
Mi késztetheti ezeket a cégeket, hogy oktatással foglalkozzanak?
Cikkünk egy konkrét példát mutat be, de természetesen többről szól, mint e kérdés megválaszolása.

Szembetűnően sok cég hirdet számítástechnikai tanfolyamokat. Úgy tűnik, manapság bárki oktathat bármit. Az oktatás színvonala mégis különbséget tesz tanfolyam és tanfolyam között. Aki tanulni akar, a felsőfokú képzéstől az 1-2 napos minimális alapozásig mindent megtalálhat a kínált tanfolyamok között.

A Műszertechnika Rt. mint a legnagyobb hazai számítógépgyártó, mindig is hangsúlyt fektetett felhasználóinak, ügyfeleinek magas színvonalú szakirányú képzésére. Úgymond hozzátartozik a cég arculatához, hogy a számítógépek és a hozzá való programcsomagok eladása mellett ezek használatának megtanulására is lehetőséget nyújt. A cégvezetés a kezdetektől vallja, hogy a korszerű ismeretanyag terjesztése alapvetően fontos: csak jól képzett — állandóan továbbképzett — szakemberek ért(ékel)hetik a mai modern programrendszereket.

Egy oktatási osztály fenntartásának szükségessége nem a nyereségben keresendő; más, nehezebben mérhető hatása van: a számítógépek alkalmazásának terjedése szorosan összefügg a gépek és a megvásárolható programok kezelésének megismerésével — vagyis az oktatással. A tanfolyamokon gyakorlottabb felhasználók találkozhatnak éppen most kezdőkkel, átadhatják tapasztalataikat egymásnak is, tanácsot tudnak adni a további vásárlásokat illetően. A hallgatók jó része a munkahelyén nem rendelkezik azokkal a programcsomagokkal, melyekkel a szakmai képzés során megismerkedik; megtetszik nekik, és utólag megvásárolják. Egy jól szervezett oktatás támogatja az új hardver- és szoftvertermékek terjesztését, végső eredményként lehetőséget nyújt a megvásárolt gépek optimális kihasználására.

A Műszertechnikánál az oktatásnak kétféle formája járatos: intenzív (5-10 napos) céltanfolyamok és szakképesítést adó (közép-, illetve felsőfokú) komplex tanfolyamok.

Amire már jogos alapozni

Az elmúlt évek során több ezer hallgatót képeztek ki ezeken a tanfolyamokon. Örömmel tapasztalták, hogy mind ők, mind a szakvizsgák független elnökei — akik gyakran konkurens oktatócégek vezetői — elismeréssel szóltak a Műszertechnikánál folyó képzésről.

A cégnek a hazai számítástechnikában betöltött szerepe, 376 fős munkatársi gárdájának felhalmozott tudása, a világ vezető számítástechnikai vállalataival (Siemens, Fujitsu, IBM stb.) tartott szoros kapcsolata kellő hátteret biztosít a tanfolyamok színvonalához. Oktatóközpontjukban IBM-kompatibilis számítógépekkel, Novell-hálózattal felszerelt termekben tartják a tanfolyamokat — mind szakmailag, mint pedagógiailag jól képzett oktatókkal. Alapos előkészítés előzi meg az oktatók kiválasztását: a jelentkezők előadói készségét, szakmai felkészültségüket próbatanításon is ellenőrzik. Ha valamelyik oktató nem kap megfelelő értékelést hallgatóitól, nem bízzák meg tovább.

A foglakozások maximum 16 fős csoportban folynak, egy gépen legfeljebb két ember tanul. Néhány vélemény szerint az eredményességet tovább fokozná, ha minden hallgatónak külön gépet lehetne biztosítani. A jelenlegi konstrukció mellett szól azonban — a dolog anyagi vonatkozásán túl — az a tény, hogy a kezdők könnyebben feltalálják magukat, ha párosával dolgoznak a számítógép mellett.

Az intenzív tanfolyamok áltagos óraszáma 30, és 5 napig tartanak. Szerveznek tanfolyamokat délelőtt és délután is. Sokan átképzésként végzik el valamelyik kurzust; egyre többen vannak a középkorosztályból. Nem kevés a munkanélküli sem, aki saját költségén „lendíti" magát tovább egy új munkalehetőség reményében.

Az angol nyelv tanítása szorosan kötődik a programok által használt terminológiához. A gyakorlatban a programok ismertetése során tanulják meg a hallgatók a szükséges fogalmakat. Az eredeti dokumentáció forgatására, használatára és megértésére is igyekeznek megtanítani mindenkit.

Speciálisat — intenzíven

Az intenzív jellegű tanfolyamokra jelentkezők többsége az IBM PC/XT-,

Számítástechnika és nyelvoktatás

Még ha el sem hagyja valaki az ország határait, hivatása kapcsán is sok külföldivel találkozik. Egyre keresettebbek a több nyelven beszélő és gépelni tudó titkárnők, üzletemberek. Az intenzív tanulás híveire gondolt a Műszertechnika, amikor felkészült az angol nyelvű számítógépes tanfolyamok meghirdetésére. Ebben a félévben a „Bevezetés a számítógépek használatába" és „Számítógépek az üzleti életben" című tanfolyamokat tartják angol nyelven. A tanfolyamokat a cambridge-i Academy of Englishsel való együttműködés fémjelzi.

AT-kompatibilis számítógépek kezelése (DOS) iránt érdeklődik: továbbra is a számítástechnikában kezdők igénylik a legtöbb segítséget. Ha valaki még soha nem dolgozott számítógéppel, akkor először egy bevezető kezelői tanfolyam elvégzését javasolják neki. Az alapok elsajátítása után jelentkezni lehet bármelyik tanfolyamra.

A speciális intenzív tanfolyamok közül évek óta nagyobb az érdeklődés a „felhasználói" jellegűek (például táblázatkezelői, szövegszerkesztői) iránt, mint az új adatbázis-kezelő eszközökkel vagy programozási nyelvekkel foglalkozókra (például Clipper, Pascal, C, Btrieve). Ez természetes, hiszen napjainkban sokkal nagyobb az alkalmazók, mint a ténylegesen programozók száma. A gyakorlott számítástechnikusok pedig sokszor autodidakta módon, egyedül tanulják meg az újdonságokat — ez persze idő kérdése. Kelendőek a hálózati tanfolyamok, a múlt évben kezdték el a WordPerfect szövegszerkesztő legújabb verziójának oktatását: kezdő és haladó szinten, kiegészítve ezzel a Műszertechnika disztribútori szerepét. Ezeken a tanfolyamokon sok a titkárnő, de nyomdász szakemberekkel is találkozhatunk. A szövegszerkesztő tanfolyamok egymást követik, és növekvőben az igény a Ventura tanfolyamokra is.

Folyamatos az érdeklődés a különleges alkalmazások iránt, így például az AutoCAD-kurzusokra főleg mérnökök jelentkeznek.

Szakképesítésért — fokozatosan

A szakképesítő tanfolyamok hetente kétszer fél nap elfoglaltságot jelentenek az igénybe vehető gép melletti és konzultációs időn kívül. A tanfolyamok összes tantárgyához jegyzetet biztosít a cég. A jelentkezők a következő bizonyítványokat kaphatják:

— személyi számítógépes szoftverüzemeltetői (1 év);
— számítógép-programozói (2 év);
— információrendszer-programozói (3 év).

Aki elvégzi a középfokú szoftverüzemeltetői tanfolyamot, a biztonságos munkához elsajátítja a számítógép és szokásos egységeinek kezelését és használatát, megismeri a legfontosabb számítógépes alkalmazási területek programjainak szolgáltatásait. Az oktatás során bemutatott programok évről évre változnak, azt a törekvést tükrözve, hogy a piacon kapható legkorszerűbb termékekkel ismertessék meg a hallgatókat. Így például ebben az évben a szövegszerkesztők között a WordStar programot — az üzleti szempontot is szem előtt tartva — lecserélték a bővebb lehetőséget nyújtó WordPerfecttel. Adatbázis-kezelést dBASE IV-gyel tanítanak, a Quattro táblázatkezelő helyett a Quattro Prót fogják oktatni a közeljövőben.

A hallgatók meg tudják oldani feladataikat számítógép-hálózatokon is. Egy-egy probléma megoldásához eszközt és programot képesek választani. Miután megismerték a legújabb és várható fejlesztési eredményeket, várhatóan jól megbirkóznak a későbbiekben felmerülő feladatokkal, könnyebben alkalmazkodnak a gyorsan változó körülményekhez.

A számítógép-programozói tanfolyam célja olyan beosztott programozók képzése, akik — vezető programozó mellett — részalkalmazások tervezésére és programozására vállalkozhatnak. Szakmai középfokú képesítést ad, kétéves. A szoftverüzemeltetői tanfolyam anyagának elvégzése után a hallgatók Turbo Pascal nyelven tanulnak programozni, majd megismerkednek a programtervezés, adatbázis-tervezés legkorszerűbb kisgépes eszközeivel. Elsajátítják az SQL adatbázis-kezelő nyelvet.

A hároméves, számítástechnikai felsőfokú képesítést nyújtó tanfolyam végbizonyítványának birtokosai már egy alkalmazás önálló megtervezésére, programozására és dokumentálására, programozói teamek vezetésére is képesek. Ezek az információrendszer-programozók a szoftverüzemeltetői és a középfokú programozási tananyag után megismerik a UNIX operációs rendszert, tanulnak rendszerszervezést, CASE-eszközöket és objektumorientált programozást.

Mindhárom tanfolyamtípuson az elméletet rendre visszacsatolja, mélyíti az életből vett feladatokon tapasztalt alkalmazás. Ezért nem különülnek el élesen az elméleti és gyakorlati órák. A lépcsőzetesség lehetőséget ad rá, hogy a hallgatók minden év végén eldönthessék, folytatják-e tanulmányaikat, vagy az adott szintű szakmai képesítővizsga letételével befejezik.

Bizonyítvány — a bizonyítványosztás után

Mindenkitől kérnek visszajelzést a tanfolyam eredményességéről. Az értékelő lap egy 12 kérdésből álló kérdőív, de helyet kapnak rajta a válaszadó javaslatai is. Ezekből a véleményekből leszűrhető, hogy a hallgatók legnagyobb problémája — nevessünk vagy sírjunk? — a büfé hiánya... Eredményességük egyik mutatójának tartják azt a jelenséget, hogy ha egy cég dolgozója elvégez egy tanfolyamot, később többen is jelentkeznek a vállalattól ugyanarra a tanfolyamra. Ebből arra lehet következtetni, hogy a megrendelők elégedettek, és belátják: több hasonló képesítésű dolgozóra is szükségük van.

Belülről nézve a gondokat: a Műszertechnika oktatóbázisának nagyon rövid időn belül le kell cserélnie kissé elavult gépparkját. A legújabb szoftvertermékek nagy része 386-os gépekre készült: legalább 1 MB központi tárigényük van. A tervek között szerepel, hogy hamarosan az e kategóriába tartozó Siemens PC-ket állítják az oktatás szolgálatába.

A honi számítástechnikai tankönyvellátásról

Ha egy programot vagy programcsomagot szeretnénk teljes egészében megismerni, elengedhetetlenül szükségesek hozzá a gyártó által szállított kézikönyvek. Ezek általában igen vaskosak, terjengősek, és idegen — leggyakrabban angol — nyelven íródtak. Az ilyen kézikönyvek kezdők oktatására alkalmatlanok, inkább referenciakönyvként szolgálhatnak a már tapasztalt számítástechnikusoknak. Egy oktatási segédlettel szemben elsődleges követelmény a tömörség, a lényeg kiemelése kezdők számára.

A megoldásokkal kiegészített feladatgyűjteményeket az emberek szinte nélkülözhetetlennek tartják az önálló gyakorlás során. Ilyen jellegű irodalomból azonban úgyszólván alig lelhető valami a könyvesboltokban. Ami van, általában nem több, mint a géppel szállított kézikönyvek fordítása, melyekre kétségkívül szükség van, de nem felelnek meg az oktatási/tanulási helyzetnek. A magyar kiadványokkal az a másik baj, hogy késnek — bár ez a lemaradás csökkent az utóbbi évben (a hatékony DTP-eszközöknek és a magánkiadások rugalmasabb bonyolításának köszönhetően).

A fenti okok miatt a Műszertechnika saját oktatási segédleteket fejlesztett ki, amelyek minőségét jelzi, hogy az ország egyik legnagyobb egyetemének innovációs kft.-je saját kiadványába is átvett — egy az egyben — bizonyos részeket az egyik jegyzetből.

Tóth Bagi Mária

„Államilag garantált" új lehetőségek

Egységes követelmények, vizsgáztatási és értékelési rend

A számítástechnikai szakképesítések megszerzésére felkészítő tanfolyami oktatás és képesítővizsgáztatás rendszerét a Központi Statisztikai Hivatal irányítja. A korábbi, ún. „szakképesítést nyújtó" tanfolyami struktúra újjászervezésére 1989-ben került sor. A központi szakképzési programokban előírt követelmények teljesítését vállaló intézményeket — amennyiben rendelkeznek a kellő személyi és tárgyi feltételekkel — a KSH Központi Számítástechnikai Szakképzési Jegyzékbe sorolja.

A számítástechnikai szaktanfolyamok koordinációjának lényege:

— szakértők széles körének bevonásával véglegesített és a szakma fejlődését, valamint a felhasználói (munkáltatói) igényeket követő központi szakképzési programokat kell teljesíteni;

— az oktatást szervező intézmények javaslatainak figyelembevételével kialakított, országosan egységes vizsgakövetelmények és vizsgáztatási, valamint értékelési rend érvényesül;

— a szaktanfolyamot szervező intézményektől független (a hallgatók érdekében az egyes helyszíneken működő), decentralizált vizsgáztatóbizottság dönt.

E három elem részletesebb bemutatását célszerűnek tartjuk. A Központi Számítástechnikai Szakképzési Jegyzékbe felvett intézmények listáját a mágneslemezen közöljük (a táblázatban megadott tanfolyam-típusszámok alapján a listában az érdeklődők arról is informálódhatnak, hogy hol, mit tanulhatnak).

A keret-tantervek az alapozó, a szakmai és a kötelezően választható tantárgyak megnevezését, valamint a vizsgaköteles tárgyak kijelölését tartalmazzák. Ez utóbbiakból — a szakmai képesítővizsgára jelentkezés előfeltételeként — akár tanfolyami hallgatóként, akár egyéni felkészülés alapján vizsgát kell tenni, vagy igazolni kell (az adott típusú tanfolyami oktatást szervező intézménynél, a korábbi tanulmányok beszámítása alapján) az ezekkel egyenértékű, három évnél nem régebbi vizsgák sikerét. A vizsgaköteles tantárgyak tartalmi elemeit is a központi szakképzési programok határozzák meg, ezek képzési céljával együtt. A keret-tantervek a szaktanfolyamok minimális oktatási óraszámait tüntetik fel. Az oktatást szervező intézmény további tárgyakat építhet be, hogy hallgatóit felkészítse a számítástechnika egy-egy konkrét felhasználói környezetben való alkalmazására is. Ezzel a lehetőséggel a velünk kapcsolatban álló intézmények közül egyre többen élnének, a többletidőnek azonban költségnövelő hatása van; ez utóbbi terheit pedig — egyéni vagy munkáltatói szinten — kevesen tudják vállalni. Az 1989-től idén novemberig befejezett összes szaktanfolyam (77) 96,1%-át olyan intézmény szervezte, amely újként kapcsolódhatott be a szakképzés rendszerébe. Némileg eltérőek, de azonos tendenciákat tükröznek a most folyamatban lévő szaktanfolyamok adatai. Ezek szerint: 9 alapfokú szaktanfolyamon 184 fő, 18 középfokú szoftverüzemeltetői szaktanfolyamon 369 fő, az összes többi szaktanfolyamon (49) 1938 fő tanul; ez utóbbiak között a Számalk 21 tanfolyammal és 1298 hallgatóval szerepel.) Országos szinten kisebb az érdeklődés a programozói, valamint szinte nincs kereslet a műszaki típusú szaktanfolyami képzés iránt. Ez igazolja a KSH korábban elvégzett elemzéseiből levont következtetéseit, hogy ti. ma Magyarországon főként a számítástechnikai eszközöket szakértelemmel alkalmazókra van igény a nemzetgazdaság minden területén.

Mindkét szaktanfolyamnál az 1 hallgató/1 gép technikai kapacitás biztosítását írják elő. (Arról, hogy ezt az előírást az oktatást szervező intézmé-

A központi szakképzési programok

Jelenleg négy szakmacsoportban (szervező, programozó, számítógép-kezelő, műszaki) tizenegyféle szakképesítés megszerzésére van lehetőség. Ezek a következők:

A tanfolyam típusszáma	A szaktanfolyamok képesítéssel azonos megnevezése	Az oktatás minimális összóraszáma	Képesítési fokozat
1.1	Számítástechnikai szervező	450	Közép
1.2	Információrendszer-szervező	800	Felső
2.1	Számítógép-programozó	450	Közép
2.1	Számítógép-programozó	800	Felső
2.21	Számítógéprendszer-programozó	850	Felső
2.22	Információrendszer-programozó	900	Felső
3.1	Személyiszámítógép-kezelő	150	Alap
3.2	Személyi számítógépes szoftverüzemeltető	350	Közép
3.3	Számítógép-kezelő (operátor)	400	Közép
3.4	Számítóközpont-üzemeltetésvezető	750	Felső
4.1	Személyi számítógépes műszaki karbantartó	400	Közép
4.2	Számítógéprendszer műszaki karbantartó	700	Felső

nyek pontosan vagy „nagyvonalúan" kezelik, eddig — már későn — csak a szakmai képesítővizsgán győződhettünk meg. A tanfolyami hallgatók érdekében a jövőben „központi vizsgabiztos"-okat kérünk fel arra, hogy esetenként — a szaktanfolyamok beindulását követően — ellenőrizzék a helyszínen az oktatás, valamint a tantárgyi vizsgáztatás körülményeit.

Sokat segíthetne a jelenlegi helyzeten, ha az átképzés (a tanfolyami szakképzés) finanszírozásához pénzügyi alapokkal rendelkező szervezetek az általuk meghirdetett pályázati kiírásokat ebből a szempontból is igényes(ebb) szakmai követelményekkel fogalmaznák meg. Azonban a regionális munkaerő-szolgálati irodák jelentős része nem ismeri még a számítástechnikai tanfolyami szakképzés rendszerét. Következésképpen nem számolhatnak azokkal a lehetőségekkel sem, amelyek ennek keretében biztosíthatók, de nem támaszthatnak megfelelő követelményeket az oktatás szervezésére pályázatot benyújtó szervezetekkel szemben sem. Ezért fordulhat elő, hogy a beérkező pályázatok elbírálásánál nem veszik figyelembe a számítástechnikai szakképzés oktatóira vonatkozó feltételeket, valamint a tanfolyam idő- és eszközigényes voltát. Gyakran azt sem mérlegelik, hogy az „eredményes tanfolyami képzés"-re elkötelezett intézmények nem (lehetnek) azonosak azokkal a szervezetekkel, amelyek „a legolcsóbb és legrövidebb tanfolyami szakképzés"-re (is) vállalkoznak. Így sem a munkanélküliként átképzésben részt vevők jogos várakozása, sem az egységes szakmai követelmények érvényesítése nem teljesülhet akkor, ha a „számítástechnikai szakképzésre" kiírt pályázat odaítélése akár egyetlen esetben is olyan szervezetet támogat, amely a Központi Szakképzési Jegyzékbe történt felvétellel együttjáró szakmai kontroll nélkül folytathatja „szakképzési" tevékenységét.

A központi szakképzési programok szakmai színvonala a KSH és az oktatást szervező intézmények együttműködésén nyugszik. A hivatal a dokumentációk aktuális változatait rendszerint az év II. negyedévében térítésmentesen küldi meg a Központi Szakképzési Jegyzékben szereplő összes intézménynek. Ennek fejében (el)várja, hogy — a képesítő vizsgabizottságok elnöki teendőinek ellátására felkért szakemberek mellett — a szaktanfolyamokat szervező intézmények is tegyenek érdemi javaslatot a szakképzés egészének vagy egyes elemeinek a korszerűsítésére. Ezek értékelését szakértő bizottságok végzik; a központi szakképzési dokumentációk fejlesztésére is így kerül sor.

E mód ellen az elmúlt időszakban mindössze egy intézmény emelt kifogást. (Ez az oktató kollektíva indokoltnak tartotta a felsőfokú információrendszer-programozó szaktanfolyam oktatási dokumentációjának megváltoztatását, azonban ennek részleteit a munkaközösség „szerzői jogának védelmében" nem kívánja megosztani a többi, hasonló képzést folytató intézménnyel; így véleményüket — sajnálattal — nélkülözni fogjuk.)

Szerte az országban — ugyanúgy

A vizsgaköteles tárgyak érdemjegyeit, valamint az alap-, közép- és felsőfokú szakképesítések megszerzését igazoló okmányok mintái a KSH-szabályzat mellékletei. (A számítástechnikai szaktanfolyamot és szakmai képesítővizsgát „legitim módon" szervező intézmények által kiadott okmányok mindegyikén szerepel a KSH engedélyszáma. Ez a szám jelent tehát garanciát arra, hogy az adott intézmény szaktanfolyama „államilag ellenőrzött" szakoktatást biztosít.)

A vizsgáztatók minősítik az oktatást szervező intézmény szakmai tevékenységét is, valamint a képzés alapjául szolgáló központi oktatási programokról is értékelést adnak. Egybehangzó észrevételeik és javaslataik alapján például napjainkra minden, 1989-ben készült központi szakképzési programot valamilyen mértékben meg kellett változtatni. A szakképzési jegyzékbe felvett intézmények között örvendetesen növekszik azok száma, amelyek alaptevékenységük szerint is oktatással foglalkoznak. Túlnyomórészt a tanfolyami szakképzésbe bekapcsolódott egyetemek, főiskolák és szakközépiskolák szakembereinek köszönhető az, hogy „a KSH központi szakképzési programjai" jelenleg országos etalonok lehetnek. Szeretném itt konkrétan megemlíteni az ELTE Általános Számítástudományi Tanszék oktatóiból alakult Informatikai Oktatási Kft.-t, a GATE Gazdaság- és Társadalomtudományi Karának gyöngyösi Vállalatgazdasági Üzemmérnöki Intézetét, a Kandó Kálmán Műszaki Főiskola Matematikai és Számítástechnikai Intézetét és az ajkai Bánki Donát Ipari Szakközépiskolát, amelyek illetékes vezetőitől és munkatársaitól mind a mai napig önzetlen szakmai segítséget kapunk.

A vizsgakövetelmények (valamint oktatási programok) teljesítése 30%-os eltérést tesz lehetővé. Az újabb anyagrészeket még az oktatás beindítása előtt, az ezekre épülő tételeket pedig a szakmai képesítővizsgáztatás tervezett időpontjának bejelentésével egyidejűleg kell egyeztetni a vizsgáztatóbizottságba a KSH által delegált szakemberekkel. A decentralizált vizsgáztatás gyakorlata kedvez a hallgatóknak: az elnök, illetve a bizottság „megy a hallgatókhoz", és nem fordítva. (A 10 főnél kisebb csoportok képesítővizsgáztatását az adott régión belül, de összevontan szervezzük.)

Tények — felnőtteknek

A két budapesti székhelyű (és a „szakképesítést nyújtó oktatás" tekintetében az 1988. év végéig monopolhelyzetet élvező) intézmény mellett mára az ország teljes területét lefedő intézményhálózat alakult ki. Mivel a konkrét szakképesítések tartalma a képzést folytató intézmény jellegétől függetlenül közel azonos, a tanfolyami és az iskolai számítástechnikai szakképzés közötti „átjárhatóság" biztosítható, és a felesleges átfedések megszűnnek.

Járulékos eredményként könyvelhetjük el, hogy a délutánra szervezett szaktanfolyamok révén a korábban kihasználatlan középiskolai, egyetemi számítógép-kapacitások is hasznosulnak.

Egyre gyakoribbak a számítástechnikai és oktatási szakintézmény által közösen szervezett szaktanfolyamok. Az így kialakuló munkakapcsolatok keretében spontán módon, a résztvevők „szabad akaratából" megvalósul az elmélet és a gyakorlat művelőinek tapasztalatcseréje (ennek hasznát a tanfolyami hallgatók is élvezik).

Az 1989 óta alap-, közép- és felsőfokú szakképesítést szerzettek (összesen 1688 fő) közül jelentős a munkanélküliként, „átképzés" céllal jelentkezettek száma (őket az állam — a képzési költségek teljes vagy jelentős hányadának átvállalásával — a regionális munkaerő-szolgálati irodák közreműködésével támogatja). Erről a területről —mint azt már az előzőekben is említettük — csak a Szakképzési Jegyzékbe felvett intézményektől származó információkkal rendelkezünk; mindezidáig nem sikerült hivatalunknak a munkaügyi irodákkal közvetlen kapcsolatot kiépíteni. Célszerűnek látszik ezt szorgalmaznunk. (Lehet, hogy kölcsönösen?!)

Bánhidai Ágnes

Valószínűségek és bizonyosságok

„Laboratóriumi" matematika

„A 70-es évek számítástechnikájának robbanásszerű fejlődését a 80-as években az oktatás reformjának kell követnie" — jósolta 1980-ban Karen Billings, a Columbia University professzornője a Microcomputing című lapban. Az előrejelzőnek igaza volt.

Az utolsó 10 évben Magyarországon is számtalan oktatóprogram, oktatórendszer készült, továbbá jó néhány világsikert aratott programcsomag van forgalomban (matematikából ilyenek például a MUMATH, a MATLAB, MATCAD).

A kezdeti oktatórendszerek elsősorban a levelező és távoktatás „gépesítését" (CMI rendszerek) tűzték célul, vagy az egyéni tanulást segítették (CAI rendszerek). Például az Illinoisi Egyetemen kifejlesztett, világszerte ismert PLATO legutóbbi verzióját olyan számítógépes oktatóhálózaton használják, amelyhez több tízezer terminál és személyi számítógép csatlakozik. A PLATO alkalmazásakor szerzett kedvező tapasztalatok mellett világossá vált: az igazán hatékony tanítási-tanulási folyamatban a legcsodálatosabb szoftver sem tudja helyettesíteni a tanárt, és igen fontos a csoportos tanulás is. A mikroszámítógépek kora lehetővé tette, hogy a számítógép jelen legyen a tantermekben, így az oktatási folyamat segédeszközévé váljék. A számítógépre alapozott oktatás (CBI: Computer Based Instruction) és a számítógéppel történő tanulás (CBL: Computer Based Learning) olyan irányzatot jelölt ki, amelyben az irányítás a tanár kezében marad, és a tanítási-tanulási folyamatban a tanuló(csoport) aktív résztvevő.

TEACHSOFT

A kellő publicitás hiánya miatt viszonylag keveset tudunk egymás munkáiról, eredményeiről. E cikk bemutat egy, a Kandó Kálmán Főiskolán használatos matematikai programcsomagot — abban a reményben, hogy sorozatot indítunk el, melyben megismerhetjük mindazokat az oktatást támogató szoftvereket, melyek ma Magyarországon „életben" vannak.

Az ötlet igen korán, 1980-ban született meg: készítsünk egy olyan programcsomagot (utólag inkább „programhalmaznak" nevezhetjük), mely a műszaki felsőoktatás matematika tananyagát dolgozza fel azzal a céllal, hogy e programokat az előadásokon elsősorban demonstrációs célokra alkalmazhassuk. A rendelkezésünkre álló eszköz az ABC80, később a HT1080Z személyi számítógép volt (emlékeztetőül: 16 kbájt memória, kazettás háttértár, gyenge grafikai lehetőségek, programnyelv: Basic). 1983-ra elkészült a TEACHSOFT első verziója, mely több mint 100 programot tartalmazott. A munkát Fenyő István és Sima Dezső vezette, a fejlesztők az ELTE, a BME és a Kandó oktatói voltak. A TEACHSOFT használatára végül csak az utóbbi helyszínen került sor. A hallgatók egy 6 db tv-készülékkel felszerelt előadóteremben követhették a programokat. 1985-ben elkezdődött a 2.verzió kidolgozása, melynek hátteréül IBM PC-ket képzeltek el.

A 2. verzió céljai, szerkezete

— Tartalmazza a műszaki felsőoktatás matematika tananyagának tipikus fejezeteit: a lineáris algebra elemeit, a numerikus sorozatokat és sorokat, az egyváltozós valós függvényeket, a differenciál- és integrálszámítást, a gyökkereső eljárásokat, a közönséges differenciálegyenleteket, a komplex algebra és analízis elemeit és a valószínűségszámítás témakörét.

— Legyen alkalmas a programcsomag az előadások illusztrálására, legyenek benne olyan példák, amelyek segítik egy-egy nehezebb fogalom, tétel, algoritmus megértését, legyen olyan bő a választék, hogy az oktató didaktikai elképzelései is érvényesülhessenek.

— Megfeleljen önálló tanulásra, tudjon lexikonként működni, és legyen felhasználóbarát: a programcsomag alkalmazása ne jelentsen semmiféle „technikai" gondot.

— Képes legyen arra, hogy a hallgató valamely, a tananyaghoz tartozó feladatot a programcsomag segítségével megoldjon.

Az egyes témakörök ún. modulokba szerveződtek, amelyek önmagukban is futhatnak, de a TEACHSOFT menüjén keresztül is elérhetők. A modulok 4-11 programjának mindegyike három részből áll: elméleti összefoglalás, demonstráló példák és szabadon választható feladatmegoldó rész. Mindháromban a képernyő utolsó sorában olvashatók a lépési lehetőségek. Természetesen az egyes programokat a felhasználó egy ajánlott sorrendben is végigkövetheti. Hasonlóképpen minden paraméter, illetve függvény megadásakor szerepel egy ajánlott érték vagy formula.

Jöjj, nézz, láss! — már tudod...

Noha a hallgatók nem „tiszta" matematikát tanulnak, viszonylag nehéz és sok matematikai fogalommal, összefüggéssel kell megismerkedniük, ráadásul rövid idő alatt. Egy-egy jó példa, érdekes ábra vagy egy célszerű numerikus táblázat sokszor többet ér, mint egy hosszú, elegáns bizonyítás. Az előadásokat a programok használata színesíti is; a kollokviumokon látható, hogy a hallgatók feldolgozzák a látottakat.

Nem túl távoli a megvalósulása, hogy a főiskolán IBM PC-kkel felszerelt matematikai laboratórium is legyen. Ebben a hallgatók a problémamegoldó készségük fejlesztéséhez olyan feladatokat kaphatnak, amelyek kapcsán kihasználhatják a programcsomagban rejlő lehetőségeket. Néhány tanulókör évente 3-4 alkalommal részt vesz ilyen jellegű foglalkozásokon. Ha a labor elkészül, akkor a programcsomag az egyéni tanulást is szolgálhatja.

Sréterné Lukács Zsuzsanna — György Anna

Intelmek a számítógéptől idegenkedőkhöz

Nem kell megijedni, nem tévesztettük el a házszámot! Nem akarjuk az ismert rossz szokás szerint azokhoz intézni a távolmaradókat elmarasztaló dörgedelmes szavakat, akik jelen vannak. Vagyis nem gondoljuk azt, hogy az alábbi levélkével az Alaplap olvasóihoz szólunk. Őket már nem kell meggyőzni a számítógép hasznáról, érdekességéről, szépségéről... De a számítógép szerelmeseinek közvetlen környezetében is mindig akadnak, akik konokul ellenállnak mindenféle kísértésnek... isten ments, nehogy a számítógép bűvkörébe sodródjanak. Mi viszont szeretnénk, ha ilyen „távolmaradók" egyre kevesebben lennének, s ebben a „beavatottak" sokat tehetnek. Az alábbi írás is a követ vájó egyetlen apró csepp lesz, ha lapunk rendszeres olvasói elolvastatják azt rokonaik, ismerőseik körében olyanokkal, akiket illet. Tehát figyelem: akikre a fenti cím nem vonatkozik, azoknak ezt az oldalt innen kezdve „ELOLVASNI TILOS, TOVÁBBADNI KÖTELEZŐ!"

Kedves Kishúgom!

Elkeseredve panaszolod, hogy az iskolában Neked is számítástechnikát kell tanulnod, pedig semmi kedved, nem érted, s nem is akarod... látod, ez elszomorít! Nézd Édesanyánkat, ki szinte ugyanezt hajtogatja, s nap mint nap püföli az öreg írógépet, miközben a szövegszerkesztő ott porosodik a szobában. Látod a kezét estére? Látod az arcát, mikor egy elrontott oldalt újra kell gépelnie?! De „nem fiam, én ahhoz nem értek..." Hát kérdem én, érted Te a színes televízió működését? Pedig minden este leülsz, s a műsort figyeled. Lám, tekintsd öcsénket, ki nemigen tudja, mely fán terem szoftver s melyen a hardver, de a játékprogramot be tudja tölteni, és igen jól szórakozik. Noha a szerkezet értése jeles erény, ám azt bátran hagyhatod a szakemberekre, a használatát tanuld meg!

Mondhatod, Anyánk is csak elboldogul valahogy. Emlékezz: az emberiség története forradalmak története. Ne csak a véres politikára gondolj, jussanak eszedbe a hétköznapok csendes forradalmai. Sok-sok évezreddel ezelőtt ember-ősünk megszelídítette a tüzet — azóta sem költözött vissza senki a barlangba télire, s a húst sem esszük többé nyersen. Vagy a tűz forradalma után jusson eszedbe a gépeké! El tudod-e képzelni mai életünket nélkülük? Ilyen földindulás zajlik ma is — az informatika forradalma. Ez a változás sem kíméletes: jaj annak, aki egyszer lemarad! Mai világunk igazi hatalmasai az információ, a tudás birtokosai.

Ne próbálj meg azzal sem védekezni, hogy nem szereted a matematikát, nem tudod a kettes számrendszert... Tudod, a szerkezet elnevezése — jelesen: számítógép — a legnagyobb becsapások egyike. Kik annak idején így keresztelték, bizony gondolkozni rest emberek voltak, s az angolok computer szavát fordították egyenest, feledvén, hogy puszta számítások végzése a gépnek csak cseppnyi erénye a tengernyi közül! Mennyivel találóbb a francia ordinateur: ez rendezőt, vezérlőt tesz magyarul. S aki csak használója az okos masinának, annak nem kell tudnia bitekről (bájtokról sem).

Sokkal inkább figyelmezd: személyi számítógép! A korábbi, többszobányi szerkentyű mára elfér zsebedben, noha tudásának növekedése éppen fordított. Annak valójában csak a beavatottak férkőzhettek a közelébe — a mait Neked tervezték. Saját munkád segítésére, saját problémáid megoldására. Te ülsz le mellé, s Te dolgozol vele. Működtető varázsszavait ma még avatott tudorai írják, de már figyelembe kell vegyék azt, hogy munkájuk általad lészen megítélve; s ha ezen munka gyümölcse, géped programja nehézkes a kezelésében vagy barátságtalan, mert logikája idegen Tőled, bizony rosszul dolgoztak! Ne szégyelld hát, ha ilyen gondjaid támadnak, akad majd mester, ki olyan cipőt készít, mely nem szorít. Ne feledd, ezen a piacon is a vevő az úr!

Kívánok hát Neked a tanuláshoz lankadatlan szorgalmat, munkádhoz erőt, egészséget; szerető bátyád,

Zoltai Péter

NETREND RT

1089 Budapest, Elnök u. 1.
Tel: 113-8217; 133-4760 • Fax: 113-9537

Alaplapok:	
AT 80286-12/16	5400
AT 286-12/16 EMS 4.0	6 400
AT 80286-16/20 EMS 4.0	7 100
AT 80286-16/26 EMS 4.0	9 900
AT 80286-16/26 NEAT, HARRIS	15 900
AT 80286-24/32 NEAT, HARRIS	18 500
AT 80386-20/26	24 600
AT 80386-25/33	25 900
AT 80386-25/43 64 kbájt cache	33 900
AT 80386-33/58 64 kbájt cache	38 500
AT 80386-40/65 64 kbájt cache	41 900
AT 80486-25/117 128 kbájt cache	76 900
AT 80486-33/150 256 kbájt cache	79 500
AT 80486-33/160	
256 kbájt cache EISA	148 000
AT 80486-50/200	
256 kbájt cache EISA	320 000
Házak:	
Baby AT + 200 W táp	5 800
Baby AT + 200 W táp DE LUXE	7 700
Normál ház és táp	8 900
Mini tower ház és táp DE LUXE	9 900
Slimline ház + táp	8 700
Slimline ház + táp II DE LUXE	9 200
Tower ház + táp	14 800
Big Tower ház + táp	19 700
101 gombos billentyű	2800
FLOPPY drive-ok:	
360 kB 5,25	5 200
1,2 MB 5,25	5 200
1,44 MB 3,5	5 200
Keret 3,5 drive-hoz	650
Monitorvezérlők:	
Monokróm	1 200
CGA	2 400
EGA	4 400
EGA és printerport	4 700
Multi EGA, 800x600	4 900
VGA 800x600	6 500
VGA 1024x768, Trident, 512 KB	7 400
VGA 1024x768, Trident, 1 MB	9 900
VGA 1024x768, Tseng, 512 KB	10 600
VGA 1024x768, Tseng, 1 MB	12 000
VGA 1024x768, Tseng, 1 MB	14 500
Grafikus processzorkártyák:	
EVER-10 TIGA (1024x768M VRAM)	66 000
A/4 monitor vezérlő	27 500
SDA-C4 1280x1024 VRAM	104 500
MDB-08 EIZO 1280x1024 VRAM	250 000
MDB-12 EIZO 1280x1024 VRAM	279 000
Koprocesszorok:	
80287-10 AMD	9 500
80287-10 INTEL	18 000
80287-10 IIT	12 000
80287-12 CYRIX	11 000
80287-12 INTEL	10 000
80287 XL INTEL	19 000
80287-20 INTEL	22 000
80287-20 CYRIX	12 900
80387-25 CYRIX	24 900
80387-25 INTEL	42 000
80387-33 CYRIX	28 000
80387-40 CYRIX	33 900
FDD-HDD kontrollerek:	
AT BUS IDE FDD/HDD	1 200
AT BUS IDE FDD/HDD	
és 1 par/ser port	1 500
AT BUS IDE FDD/HDD	
és 1 par/2ser port	1 899
AT BUS IDE FDD/HDD	
és 1 par/2ser 1 g.	2 100
IDE FDD/HDD saját BIOS preformat	3 900
WD 1007 komp. ESDI kontr.	16 500
XT FDD kontroller	2 300
AT/XT FDD kontroller	3 500
XT multi I/O kártya	2 750
XT HDD kontroller	3 900
AT HDD/FDD kontroller	
WD/1006 komp.	5 000
E-6295 ESDI EISA kontr.,	
2 MB CACHE	72 000
E-6295 ESDI EISA kontr.,	
4 MB CACHE	98 500
S-3290 SCSI EISA kontr.,	
4 MB CACHE	96 000
IDE winchesterek:	
ST 351, 28 msec, 40 MB	15 300
ST-157A, 28 msec, 40 MB	14 900
WD 140, 12.8 msec, 40 MB	24 900
ST-1102A, 19 msec, 89 MB	24 500
ST-3096A, 15 msec, 84 MB	28 900
ST-1144A, 19 msec, 125 MB	30 600
ST-3144A, 16 msec, 125 MB	35 900
ST-1239A, 15 msec, 211 MB	52 900
MFM winchesterek:	
ST-225 65 msec 22 MB	13 600
ST-251-1 28msec 40MB	22 100
CDC 4097 28 msec 80 MB	49 900
ESDI winchesterek:	
CDC 4182E 19 msec 160 MB	76 900
CDC 4383E 16 msec 338 MB	104 900
CDC 4384E 14 msec 338 MB	118 500
CDC 4442E 16 msec 380 MB	149 900
CDC 4766E 16 msec 676 MB	159 900
SCSI winchesterek:	
CDC 2383N 14 msec 338 MB	118 500
CDC 4766N 16 msec 680 MB	159 900
CDC 41200 16 msec 1050 MB	269 000
Monitorok:	
Egyszínű (borostyánsárga)	8 500
Egyszínű (borostyánsárga), Dualsync	8 700
Egyszínű (papírfehér)	9 500
EGA	24 000
EGA	26 800
EGA	27 900
Mono VGA (640x480)	13 900
Mono VGA (1024x768)	19 600
Mono VGA, fehér, A/4	29 900
VGA (640x480)	27 900
VGA (1024x768)	29 600
VGA Multisync (1024x768)	39 900
VGA Multisync (1024x768),	
non interl.	83 500
EIZO 9060 (1024x768)	88 000
EIZO 9070 (1024x768)	148 000
EIZO 9400 (1280x1024)	271 000
EIZO 9500 (1024x768)	286 000
NEC 3D (1024x768)	82 000
NEC 3D SSI (1024x768)	86 000
NEC 4D (1024x768)	147 000
NEC 4D SSI (1024x768)	158 000
NEC 5D (1280x1024)	259 000
NEC 5D SSI (1280x1024)	299 000
NEC Monograph System	
(1024x1024)	119 000
CC 9511 (1280x1024)	158 000
Tatung (1280x1024)	175 000
RAM IC-k, RAM modulok:	
4164-1064kB	135
4464-08 4x64 kB	300
41256-12 256 kB	155
41256-10 256 kB	165
41256-08 256 kB	170
41256-07 256 kB	195
41256-06 256 kB	220
44256-10 4x256 kB	450
44256-08 4x256 kB	500
44256-07 4x256 kB	600
411000-08 1MB	520
SIMM	1 250
SIP	1 300
SIMM	4 200
SIMM	4 400
SIP	4 500
SIMM	19 900
SIMM	21 000
STERAMEREK:	
Colorado 120MB internal	29 900
External kit	14 900
Streamer kazetta	4 700
Archive FT60 external	44 900
FT 60 kontroller	15 000
Hálózati eszközök, hálózati csatolók:	
ARCNET 8 bites	3 900
ARCNET 16 bites	6 900
8 bit NE-1000 komp.	8 900
16 bites NE-2000 komp.	10 500
16 bites ZOT (NE-2000 komp.)	11 500
NE-3200, EISA, 32 bites	9 400
DE-100 Ethernet 8 bit	14 000
DE-150 Ethernet 8 bit	14 500
DE-200 Ethernet 16 bit	16 000
WD-8003 Ethernet 8 bit	15 500
WD-8013 Ethernet 16 bit	18 000
Ethernet és ARCnet-kiegészítők:	
2 portos repeater, transceiverrel	39 900
DE-802 2 portos repeater,	
transceiver-rel	99 000
DE-804 4 portos repeater,	
transceiver-rel	148 000
BOOT-EPROM ARCNET-hez	2 000
BOOT-EPROM	
NE1000/NE-2000-hez	2 500
BOOT-EPROM D-Link-hez	3 000
Fali csatlakozó (BNC)	1 000
93 Ohmos lezáró	270
50 Ohmos lezáró	300
BNC T-dugó	360
ARCNET passzív hub	1 000
ARCNET aktív hub-8 external	11 500
ARCNET aktív hub-4, internal	5 800
ARCNET aktív hub-4 + 1 internal	6 900
BNC aljzat, BNC dugó	245
Kábelek:	
50 és 90 Ohmos	90
Modemek és FAX-kártyák:	
2400 baud belső	8 900
2400 baud external	12 900
2400 baud external, MNP-5	15 000
9600 baud EXTERNAL, MNP-5	48 500
FAX kártya/2400 baud	22 000
Hálózati FAX, 8 felhasználós	72 000

Speciális Ethernet-eszközök, fénykábelek, fénykábeles kiegészítők.
Irodabútorok, fénymásolók, telefaxok nagy választékban.
Egyedi és eseti javítások, hálózati installálások.
Pocket és UTP-hálózati csatolók igény szerinti szállításal!
És még sok minden egyéb...
Áraink csak készpénzfizetésre, illetve a visszaigazolt megrendelő alapján, egyszámlánkra előre utalásra értendők, és telephelyünkön történő átvételre vonatkoznak!
Külön szerződés alapján végzünk üzembe helyezést a megrendelő telephelyén!
A Netrend Rt. a Novell Inc., a CADKEY, valamint a Santa Cruz Operation hivatalos dealere.

INFORMÁCIÓKÉRÉS: 03 ▶

A magyar helyesírás-ellenőrző programokhoz

Tes(z)tre szabott gondolatok

Az eddig elkészült helyesírás-ellenőrző programok lényegében azon az elven működnek, hogy egy adott nyelv szókincse a nyelv helyesírási szabályrendszerének megfelelően egy jól működő programban van rögzítve, s a program a rögzített és a felhasználó által írt szóalakokat hasonlítja össze futása közben. Ha a kettő megegyezik, a gép elfogadja a beírt szót, ha a felhasználó által leírt szóalak nem szerepel a programban szereplők között, a gép helytelennek minősíti azt.

Mi szerepeljen a szókincsben?

Az elvből az következik, hogy a felhasználó csak akkor tud jó hatásfokkal és kényelmesen használni egy helyesírási programot, ha az általa leírt szóalakok — természetesen csakis helyes írásképpel — mind szerepelnek az adatbázisban. Így a program nem utasítja el a helyesen írt nyelvi formákat, de jelzi a helyteleneket, hibásakat. A szókincsnek tehát tartalmaznia kell a közszavakat, s az igényesebb felhasználók érdekében az ún. műveltségi szavakat (ezek legtöbbje valójában idegen szó: imprimatúra, moralitás, neutralizáció stb.). Félrevezető lehet viszont, ha teljesen elavult, a mai nyelvhasználatban már nem élő, illetve ha nyelvjárási szavakat tartalmaz. Zavaró tehát olyan szavakat szerepeltetni benne, mint például: üteny, csömbő, homp, csorvasz, járlat, farcsik stb.

Kissé más a helyzet a szakszavakkal. Közülük azokat, amelyek a köznapi nyelvhasználatban gyakoriak (szénmonoxid, desztillál, terápia, denaturált stb.), érdemes szerepeltetni az adatbázisban. Arra viszont, hogy akárcsak a középiskolai tanulmányok során elsajátítandó szakszókincset is tartalmazza a program, egyetlen szerző sem vállalkozhat. Ezen szakszavak száma ugyanis olyan nagy, hogy talán felül is múlná a közszavakét. E kérdés megoldásában más utat kell járni. Fel kell majd dolgozni a jövőben az egyes szakterületek szókincsét a speciális felhasználók, a szakterület művelői számára, akár külön programként, akár a köznyelvi alapprogram kiegészítéseként.

Nehéz feladat a szerzők számára a tulajdonnevek kezelése. A fő kérdés, hogy a sok millió tulajdonnév közül melyeket tartalmazza a program. Nyilvánvaló, hogy akármilyen sok tulajdonnevet is tartalmaz, akkor is számosat hiányolni fog minden felhasználó. Ha pedig a hibalistán így is, úgy is szerepelni fognak tulajdonnevek, nem nagy kényelmetlenség, ha az írásmű szerzője néhánnyal többet ellenőriz. A legjobb megoldásnak az kívánkozik, hogy a felhasználók valamilyen módon állandóan bővíthetik a programban szereplő tulajdonnevek körét; így elérhető, hogy mindenki a saját igényeihez igazíthassa a saját ellenőrző programját.

Az alaktani változatok

Nagyon fontos kérdés, hogy milyen és melyik alaktani változatát szerepeltessük egy-egy szónak. Naszódi Mátyás lapunk 1991/5. számában alaposan tárgyalta ezt a kérdést. Mivel azonban egy magyar „spelling checker" minősége, tehát használhatósága döntő mértékben a toldalékos (ragos, képzős, jeles) szóalakok helyes kezelésén múlik, érdemes néhány problémára ismételten és részletesebben rávilágítani. Nem véletlen, hogy nem született már korábban magyar helyesírás-ellenőrző program. Ugyanis míg egy angol nyelvű spelling checker 50 ezer alapszó esetén kb. 100 ezer szóalakváltozatot tartalmaz, ugyanennyi magyar alapszóból a különböző képzők, ragok, jelek, ill. ezek kombinációinak alkalmazásával több milliárd szóalakváltozatot hozhatunk létre. Azt talán nem is kell külön bizonygatni, hogy ekkora szóadatmennyiség helyes kezelése a korábbi, hagyományos, lassan már közismert spelling checker-készítési eljárásokkal lehetetlen. Ezért is írta e kérdéskört elemezve Naszódi Mátyás: „A feladat tehát valóban nagyon összetett, és matematikailag is definiálatlan, ezért voltaképpen megoldhatatlan." (Uo. 44. o.) Igaza is van, azzal a megszorítással, hogy az ismert, hagyományos módszerekkel valóban megoldhatatlan. S hozzá kell még tenni: a magyar nyelv minden eddiginél gazdagabb és főleg pontosabb alaktani leírásának birtokában szabad csak vállalkozni egy magyar program elkészítésére.

Alaktani gazdagságunk

Sokan kétségbe vonják, hogy valóban ilyen gazdag-e a magyar nyelv az alaktani változataiban. Nos, a kétkedőknek néhány adat. Egy magyar főnévnek 42 nominativusi alakváltozata lehet. Az indoeurópai nyelvekben meglevő egyes és többes számú alanyeseten kívül ugyanis még 40 birtokos személyragos, birtokjeles alakváltozat él a magyar nyelvben (barátom, barátaid, baráté, barátaidé stb.). Ezek mindegyike lehet a mondatban alany vagy állítmány, s mindegyikükhöz hozzájárulhat a tárgy vagy bármelyik határozó ragja (barátomét, barátaiddal stb.). Így egy főnévnek közel ezer, ragos alakváltozata is lehet — s hol vannak még a képzők? Vannak ugyanis olyan főneveink — nem is kevesen —, amelyekhez néhány tucat képző is hozzátehető, s nagyon kevés azoknak a száma, amelyek csak egy-két, illetve egyetlenegy képzővel sem szerepelhetnek. Sőt, egy képző után további képzőt is használhatunk, azaz egy szónak nemcsak számos képzett, hanem továbbképzett alakja is él nyelvünkben. Az sem szokatlan, hogy a továbbképzés során ugyanazt a képzőt ismételten használjuk (igazIságIosIság). Nem meglepő tehát, hogy egy-egy alapszónak néha több százezer származékát

is szerepeltetni kell a programban. Ehhez gyakorlatilag új nyelvészeti alapkutatások szükségesek.

Nem véletlen, hogy az első magyar helyesírás-ellenőrző program, a NyelvÉsz nyelvészeti előmunkálatai több mint két évig tartottak. A magyar nyelvészeti szakirodalom ugyanis nagyon sok összefüggést tárgyal, csak épp az hiányzik belőle, amire a programok szerzőinek leginkább szükségük van: minden egyes szó minden egyes alaktani változatának pontos leírása.

A szótövek

Egyáltalán nem mindegy, hogyan építjük fel a programnak azt a részét, amely a toldalékos alakokat kezeli. A magyar nyelv ugyanis nemcsak azért páratlan alaktanának gazdagságában, mert sok toldalékos alakja van egy szónak. További két jelenség növeli e gazdagságot. Az egyik a szótövek rendszere. Szavaink egy részéhez úgy kapcsolhatók a toldalékok, hogy az alapszó mindig változatlan marad.

Más szavak esetében a különböző toldalékok különböző, egymástól néha jelentősen eltérő tövekhez járulhatnak. Nyelvünkben igen sok többtövű szó van (bokor — bokrot, ló — lovat stb.), ezek jelentős részét szinte minden közlendőben használjuk. Az sem ritka, hogy egy szónak három, négy vagy akár több töve is él, gondoljunk csak a létigére (van|nak, vagy|ok, volt|ál, len|nél, legy|él stb.). Emiatt minden egyes többtövű szó esetében meg kell állapítani, hogy melyik toldalék melyik tőhöz tehető hozzá. Miután a különböző többtövű szavak sokféleképpen toldalékolódnak, e jelenség pontos feltárása elengedhetetlen, különben a program lehetetlen, nem is létező szóalakokat is „ismerni" fog (tehent, jóbb, kétté, morogó stb.). Annál is inkább fontos ez, mert előfordul, hogy egy ragozási soron (paradigmán) belül is váltakoznak a tövek (mozl|gok, mozog|sz, mozg|unk, mozog|tok stb.). Olyan szavaink is vannak, amelyek mindkét tövéhez hozzátehető ugyanaz a toldalék, de eltérő alaktani formájában (sátor|t, sátr|at stb.).

Ha például az egyik leggyakoribb képzőnk használatát, az igéhez kapcsolható „-ás" vagy „-és" képzőét akarjuk rendszerbe foglalni, először ki kell válogatnunk azokat az igéket, amelyekhez e képzők nem tehetők hozzá (nincs ugyanis van|ás, hallatsz|ás, látsz|ás stb. szóalakunk). Amely igékhez hozzátehetők, azokat először hangrendi szempontból kell szétválogatnunk (a mély hangrendűekhez az „-ás", a magasakhoz az „-és" járul). Ezután a többtövű igék mindegyike esetében el kell döntenünk, melyik tőhöz ragasztható hozzá a képző. Az eredmény eléggé meglepő: csak e képzőpár szempontjából több tucat csoportra osztandók a magyar igék.

A toldalékok

A másik jelenség, amely az alaktani gazdagságot növeli, az a jól ismert tény, hogy a magyar szavak toldalékainak, a képzőknek, a ragoknak és a jeleknek az esetek többségében több, hangtanilag eltérő változata van (pl.: tábl|án, asztal|on, szék|en, könyv|ön). Látszólag egyszerűnek tűnik a dolog: a magas hangrendű szóhoz magas hangrendű toldalék járulhat (szekrény|ben), a mélyhez mély (szobá|ban). Csakhogy a szavak egy részének hangrendi mivolta nem a bennük levő magánhangzóktól, hanem egy sajátos történelmi hagyománytól függ. Az „iszik, bír, híd, cél" stb. szavak mély hangrendűek, de a „kódex" például magas. Számos olyan szavunk is van, amelyek után mind a magas, mind a mély toldalékok is használhatók (fotel|ben, fotel|ban stb.). Akad olyan is, amelynek bizonyos toldalékos alakjai csak mély, mások magas hangrendűek is lehetnek (derek|as; de: derék|ban. derék|ben).

Külön érdekesség, hogy a korábban vázolt 42 nominativusi alaknak megfelelően egy szó különböző alakváltozataihoz ugyanazon rag eltérő formájában kapcsolódhat. A „tömb" szó tárgyragos alakja: „tömb|öt", a „tömb|öm" ragos alakban ugyan két „ö" hang van, de a tárgyragos alakja: „tömb|öm|et". Az egyes szám, harmadik személyű birtokos személyrag és a birtokjel után a rag csak a „-t": „tömb|jé|t, tömb|é|t". Az már csak apróságnak tűnik, hogy ez a birtokos személyrag hangváltó, ún. relatív tő (asztal|a; de: asztal|á|n), ám bizonyos toldalékok előtt nincs hangzóváltás: „mozgás|a|kor" (de: mozgás|á|ban).

A szókincs alaktani feldolgozása

Nyilvánvaló, hogy egy helyesírás-ellenőrző program csak akkor lehet megfelelő, ha ezt az alaktani tarkaságot jól kezeli. Minden egyes szó esetében meg kell állapítani a tövet (a töveket), s hogy melyik toldalék tehető utána (utánuk), melyik formában. A feldolgozás során a szavakat külön-külön, egyenként kell vizsgálni. Az adott szó szótári, vagyis alapalakjából ugyanis nagyon sokszor nem következik automatikusan, melyek a megfelelő, helyes toldalékos alakváltozatai.

Gömbök — földek

Egy egyszerű példa: A „gömb „ és a „föld" szó hangtanilag azonos felépítésű (egy mássalhangzó, ugyanaz a rövid magánhangzó, két mássalhangzó). Bizonyos toldalékos alakjaik azonosak (gömb|höz, föld|höz; gömb|ön, föld|ön), mások eltérőek (gömb|ök, gömb|öt; föld|ek, föld|et). A szó alaktani felépítése, azaz a hangteste tehát nem igazít el a lehetséges toldalékok meghatározásában és használatában. A toldalékolás szempontjából csak ezek, az ilyen hangtestű szavak több mint tíz különböző csoportot alkotnak.

Miért fontos a pontos alaktani leírás?

Bármilyen nyelvű, de jó spelling checker használatával csupán annyit érhetünk el, hogy nem írhatunk le olyan szóalakot, amely az adott nyelv szabályai szerint nem létezik, vagyis nem helyes. Az „értelmes" hibát (Tarolja a búzát. — Tárolja a búzát.) egyetlen program sem tudja megakadályozni. Az ilyen típusú értelemzavaró hibák (az ún. „leiterjakab"-ok) kiküszöbölése továbbra is a szerző, az olvasószerkesztő vagy a lektor feladata marad.

Sajnos az egyszerű elütésekből származó hibák még a gyakorlott gépírók munkáiban is gyakran előfordulnak, a számítógép felhasználásával készített napilapjainkban pedig, mint az közismert, mindennaposak.

Volnék — „volnák"

Ha egy magyar helyesírás-ellenőrző program összeállítói nem kezelik igen pontosan és hatékonyan a magyar igeragozási rendszert, akkor egy szöveg ellenőriztetése során igen furcsa, a nyelvben nem is létező formák csúszhatnak át a szűrésen. Igaz, hogy pl. a „legyétek" vagy a „volták" „igealakokat" épeszű, magyar anyanyelvű ember nemigen fogja leírni, viszont elég gyakori leütési hiba, hogy „e" helyett „é"-t, „a" helyett „á"-t ütünk. Az is meglehetősen szokványos, hogy a kívánt billentyű helyett a szomszédját ütjük le. Fennáll az a lehetőség, hogy pl. az „é" helyett „á" kerüljön a leírt szóba. A helyesírás-ellenőrzőjében bízó szerző nem is veszi észre, hogy a helyes „volnék" a helytelen „volnák" formában maradt a szö-

vegében. Ha a program megengedi a tárgyatlan igék tárgyas ragokkal való ellátását, ezt a hibát, illetve az ilyen típusúakat nem fogja jelezni.

A „volnák" esetében ez annál is kínosabb, mert az ún. „nákolás" tipikus nyelvhelyességi hiba. A nyelvhasználók bizonyos rétege a „borotválkoznék, tudnék, akarnék" stb. igealakok helyes „-nék" ragját a tárgyas ragozás többes számú, harmadik személyű (ott helyes!) „-nák" ragjával helyettesíti még akkor is, ha az igének csak alanyi ragozású formái léteznek a magyar köznyelvben. A „volnák" tehát csak mint a nyelvhelyességileg igencsak kifogásolható, helytelen alak létezik — leírása vagy szóbeli említése durva hibának minősül, használójára azonnal rásütik a műveletlenség bélyegét (holott csak egy egyszerű melléütésről van szó, amit a program a nem megfelelő nyelvi felépítése miatt nem tud kiszűrni).

Ha a tárgyas és a tárgyatlan igék ragozásbeli különbségét nem tartalmazza pontosan a nyelvi leírás, a véletlen elütésből eredő hibák, amelyeket a program nem szűr ki, akár komikus hatást is kelthetnek (lehetem, leheted, leheti, kelljed, kelljük stb.).

„Rakjék" — „kapaszkodják"

Hasonló a hibalehetőség, ha nem teszünk különbséget az ikes és a nem ikes igék között. Ez esetben programunk a helyes, ikes ragozású „kapaszkodjék, fogódzkodjék" stb. mintájára helyesnek fogja minősíteni az „olvassék, rakjék" stb. nem létező alakzatokat is. A tárgyas raggal helytelenül ellátott „kapaszkodják" stb. típusú formák meglétével együtt ez azt jelenti, hogy a program bizonyos esetekben nem kezeli az „á" és az „é" közötti különbséget. A két betű pedig a legtöbb klaviatúrán egymás mellett helyezkedik el, s mi védhet jobban a melléütés következményétől, mint a jó helyesírás-ellenőrző?

Meghal — meghall

Az egy „l"-lel írandó hal (halni) és a kettővel írandó hall (hallani) igék közötti különbségtevés már az általános iskola alsó tagozatos diákjaitól is elvárható. Egyes toldalékos alakjaiban a két ige írásképe csak az „l" betűk számában tér el egymástól: (meg)halok; (meg)hallok.

Más alakok esetén, például a múlt idejűekben a különbség szignifikáns: valaki meghalt valamilyen betegségben; valaki meghallott valamit.

Ha a program által működtetett nyelvi leírás nem teljesen pontos, előfordulhat, hogy a billentyűket rosszul nyomva a „hallnék, halltam, halottál, halottatok" stb. helytelen formákat írja le valaki akaratlanul is a szövegében.

„Alat" — „építet"

Komoly nehézséget okozhatnak az ige műveltető képzői: „-at, -et, -tat, -tet". Ha a nyelvtani leírás nem elég finom felbontású, azaz nem tesz különbséget közöttük (pl.az „épít" után megengedi az „-et" képzőt is), számíthatunk arra, hogy számos elütésünket a gép helyesnek fogja minősíteni. Az „alatt" névutó egy „t"-vel is „helyes" lesz, hiszen az „altat" helyett az „alat" is „jó" az alszik ige műveltető formájaként. Ugyanakkor az „épít" ige több múlt idejű alakja jelenhet meg tévesen („építet, építetem" stb.).

Mit teszteljünk?

Megengedhetetlen, hogy a helyesírás-ellenőrző olyan, alaktanilag helytelen szóalakokat is jónak minősítsen, amelyek egyszerű elütéssel is megjelenhetnek.

Ha az ellenőrzés nem veszi észre a betűkihagyást, a program használhatósága kérdésessé válik.

Ha a hangzóvesztő szavak nyelvi leírása nem kielégítő, furcsa szóformákat írhatunk le egyszerűen úgy, hogy tévedésből egy betűt nem ütöttünk le (dolgzom, dolgztok, dolgzzak stb.).

Nagyon fontos, hogy jól kezeljük az ún. hasonulásokat. Ha a program elfogadja a hasonulás nélküli formákat is (keresje, vehesje, olvasja), kisebb baj, mintha eleve rossz formákat tartalmaz („csehhel, juhhal" stb. helyesen: csehvel, juhval).

Számos hiba forrása lehet, ha a szavakat a program készítői már az adatbevitel során helytelenül rögzítették. Például: ha az „-ít" képzős igék rövid „i"-vel is leírhatók (mentesit, teljesit, törvényesit stb.), a program végleg nem tudja teljesíteni feladatát, hiszen ez esetben nem egyszerű elütésről, hanem valódi helyesírási hibáról van szó! Rossz nyelvi leírás következménye lehet, ha a magas hangrendű, mássalhangzóra végződő melléknevek középfokú alakjából hiányozhat az egyik „b" („vöröseb, keserveseb, merészeb" stb.).

Az ilyen jellegű hibalehetőségek megszüntetése már csak azért is fontos, mert félő, hogy az elkényelmesedett felhasználó — az általa használt programban bízva — kényelmetlenül érezné magát, amikor nyomtatásban látja azt az elütésből származó, hibásan írt szót, amelynek helyes írásképét különben soha sem téveszti el.

A tesztelés alapja

A helyesírás-ellenőrző programokat „A magyar helyesírás szabályai" (11. kiadás) alapján kell elkészíteni. Ha egy program olyan helytelen szóalakokat tartalmaz (pl.: a helyes „higgye" helyett: „higye"), amelyeket a szabályzat tilt, érdemlegesen nem használható.

Természetesen a helyesírási szabályzat nem tartalmazza az összes szó összes toldalékos alakját, de sok összefüggésre világít rá, sok eligazítást találunk benne. Ha egy vitatott szóalakról nem tudunk a szabályzat alapján dönteni, jelentős magyar nyelvészeti szakirodalom áll rendelkezésünkre az alaktani változatok helyességének kérdésében is.

Mi rosszabb?

A működést háromféle hiba zavarhatja.

1. A programban nem szerepel a kérdéses szó.

Ezt a hibalehetőséget soha nem lehet megszüntetni (néhány százezer alapszó rögzítésével sem!), de a szókincset bővítve a gyakoriságát csökkenthetjük.

2. A program rossznak minősíti a helyes szóalakot.

Ilyenkor a felhasználó felülbírálhatja a programját, ha tudja, melyik a kifogástalan íráskép. Ha nem tudja, a szövege hibás maradhat.

Természetesen a felhasználó számára ezek a hibák meglehetősen bosszantók, s csakhamar nem fogja használni a programot.

3. A program helyesnek minősíti a rosszul írt szót.

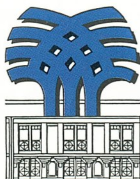
Ez a legsúlyosabb hiba, hiszen pontosan azért használják, hogy ne engedje meg a helyesírási hibákat. Meggyőződésem, hogy az utóbbi két hibalehetőség kiküszöbölhető. Ez már csak azért is szükséges, mert a helyesírás-ellenőrző program csak akkor használható, ha a magyar szavak lehető legtöbb alakváltozatát tartalmazza.

De csak a helyeseket!

Seregy Lajos

NAPI DÍJAT A CÉDRUSTÓL!

Naponta számítógépsorsolás az áruházban



Cédrus Karolina Áruház

Budapest XI., Karolina út 17.

Nyitva: 8.30 - 18.30

Szombaton: 8.30 - 13.00



INFORMÁCIÓKÉRÉS: 29 ▶

DOS-kiterjesztők

A Phar Lap DOS Extender

Hogyan lépjük át a jó öreg DOS korlátait? Ezt a kérdést minden programfejlesztő kénytelen-kelletlen felteszi magának egy idő után, amikor minduntalan ezeknek ütközik. S minthogy a DOS közel egy évtizedes élete során a verziószámok folyamatos növekedése ellenére sem sokat fejlődött (fogalmazhatnánk úgy is, hogy megőrizte karakterét), gazdagon burjánzik azoknak a kiegészítő termékeknek a piaca, amelyek gyengeségeit hivatottak ellensúlyozni.

A szűkös DOS

A DOS-nak két alapvető hibáját kell megemlíteni: a memóriakezelést és a taszkkezelést. Eredendő bűne abban rejlik, hogy az egyre intelligensebb INTEL 80286/80386/80486-os processzorok ún. valós módjában fut. A kompatibilitás szándéka derék ugyan, de a sebességnövekedésen kívül a processzorok más jó tulajdonságát nem kamatoztatja. Képletesen szólva egy 486-os AT a DOS operációs rendszer alatt olyan, mint egy Porsche motor Trabant karosszériába építve. A 8086/8088-as Intel processzor 24 bites címzésével csak 1 Mbájt RAM elérésére alkalmas. Ezt az örökséget nyögik ugyan a fejlettebb társai, de már a 80286-os processzor is 16 Mbájt címzésére alkalmas, hardverből támogatott ún. szegmentált memóriakezeléssel rendelkezik, nem is beszélve a 386/486-os 4 Gbájtig használható, szegmentált lapozásos technikájáról. A DOS-t a változások szele kezdetben csak meglegyintette: az 1 Mbájt feletti memóriát RAM diszkként lehetett használni, de a futtatható programok mérete továbbra is korlátozott maradt: a ROM területeket és a video-RAM-ot leszámítva az alsó 640 kbájtból (A0000H bájtból) csak akkora memória állt rendelkezésre, amekkorát a DOS rendszerprogramjai, készülékmeghajtói és egyéb rezidens (TSR) programok szabadon hagytak, de a legjobb esetben sem több, mint 600 kbájt. Így azután nem kellett sokáig várni, amíg megszülettek az első testesebb, illetve nagy adattömböket kezelő programok, amelyek betöltés után „Insufficent memory!" hibajelzéssel okoztak múlhatatlan pillanatokat szerzőiknek.



A szegényebbek visszafogták magukat mind az adatok, mind a programok terén, esetleg kínok kínjával overlay struktúrába szervezték programjaikat, a tehetősebbek LIM (Lotus — Intel — Microsoft) EMS (Expanded Memory Specification) specifikáció alapján készült memóriakezelő kártyát vásároltak (amely viszont csak az ezt a szabványt támogató szoftverekkel volt hajlandó együttműködni).

Mi volt a megoldás?

A 386-os processzor elterjedésével megnyílt a lehetőség a különböző késleltetett lapozási rendszerek térnyerésére (demand paging), amelyeken az ún. 'expanded' memória fogalma alapul. Ezeknél a rendszereknél egy szoftver segítségével érhető el az, amiről korábban a hardver gondoskodott. A végrehajtandó program egy része, vagy akár teljes egésze az 1 Mbájt feletti címtartományban található, és csak az éppen végrehajtás alatt álló programrészletet tartalmazó ún. lap "töltődik le" a DOS által is megcímezhető tartományba. Az idézőjel arra utal, hogy az adatok fizikailag nem mozognak, csak a processzor transzformálja a címeket.

Ezt a módszert használja a DOS 5.0 is, ahol többek között a különböző készülékmeghajtók és TSR programok kerülhetnek a 640 kbájt és 1 Mbájt közötti címtartományba, valamint az 1 Mbájt feletti tartomány első 64 kbájtos szegmensébe. A memóriakezelés hiányosságainak áthidalására egy sor megoldás született: AboveMEM, Turbo EMS, 386MAX, QEMM-386. Ebből a sorból némileg ki-

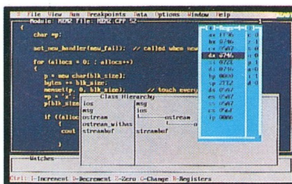
lóg a Quarterdeck DESQview, illetve a Microsoft Windows programja, mert a memóriakezelésen túl megoldották a DOS másik nagy hiányosságát is: ezekben már a taszkok egymással párhuzamosan is futtathatók.

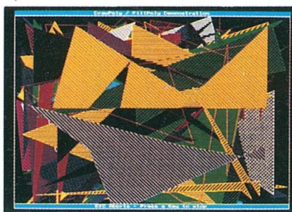
A Windows egy más programozási környezetet jelent, ugyanis a speciálisan Windows alá fejlesztett programok csak Windows alatt futnak. A DESQview vagy más multitaszkos rendszerek (mint például a VM386) viszont nem változtattak a DOS azon dogmáján, hogy csak 640 kbájtot tud megcímezni. Ennek megoldására fejlesztették ki az ún. DOS Extendereket, amelyek nagyobb memóriaigényű programok futtatását is lehetővé teszik. (Így készült el például a Lotus 1-2-3 Release 3 a Rational Systems DOS/16M rendszerével, vagy a Borland Paradox 386, illetve az AutoCAD 386 a Phar Lap 386|DOS Extender segítségével.) Ha a felhasználó ezeket a termékeket futtatja, nem is veszi észre, hogy itt 640 kbájtot meghaladó programokról van szó.

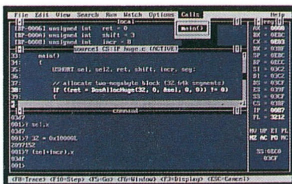
A jól bevált szoftveres gyakorlatot követve, a memóriakezelőket, DOS Extendereket előállító cégek mindössze önmagukkal igyekeztek kompatibilisek maradni, így minden felhasználónak előre el kellett döntenie, melyik DOS Extendert alkalmazza. Ezen az áldatlan állapoton segített a szabványosítás, ahol a nagyobb és ügyesebb hal lenyelte a kisebbet. A Phar Lap és Quarterdeck cégek 1989-ben lefektették az ún. VCPI (Virtual Control Program Interface) szabványt, ahol a többiek (Autodesk, Lotus, Quadram, Qualitas, Rational Systems) csak támogatóként vettek részt. Nem egészen egy év múlva megszületett a DPMI (DOS Protected Mode Interface) szabvány, ahol a résztvevők köre már sokkal tágabb: Borland, Eclipse, IGC, Intel, Locus, Lotus, Microsoft, Phar Lap, Phoenix Technology, Quarterdeck és Rational Systems. Ez a felállás már elegendő garanciának tűnik arra, hogy az ezután készülő programok döntő többsége eleget fog tenni a DPMI memóriakezelési szabványnak.

„Kiengedett" öltözetben

A Phar Lap DOS Extender támogatja az XMS, VCPI és DPMI szabványok mindegyikét. A DPMI-támogatás lehetővé teszi, hogy egyes Extended DOS alkalmazások a DOS, illetve a multitaszkos operációs rendszerek (mint Windows, OS/2, DESQview 386, UNIX 386 és VM/386) további változatai alatt futhassanak. Jelenleg a DOS mellett csak a DESQview 386 és a







Windows felel meg ezeknek a feltételeknek (igaz ugyan, hogy utóbbinak mindhárom módjában: real, standard és enhanced). A VCPI szabvány további támogatása viszont a fentiekhez képest még más előnyöket is biztosít: lehetővé teszi EMS emulátorok együttműködését a Phar Lap DOS Extenderrel fejlesztett Extended DOS-alkalmazásokkal, sőt az utóbbiak párhuzamos futtatását is a DESQview 386 multitaszkos környezetében.

A Phar Lap 286|DOS Extendere tulajdonképpen a 80286-os processzor védett (ún. protected) módjában futtatja le az általunk megírt programot, amely ily módon egészen a megcímezhető 16 Mbájt memóriakapacitás erejéig terjeszkedhet. Ehhez nem kell semmit sem változtatnunk a Microsoft C-ben vagy FORTRAN-ban készült forrásszövegen, csak újra kell szerkesztenünk a programot a '-Lp' opcióval:

```
C> CL -AL -Lp bigprog.c
C> RUN286 bigprog
```

Itt „RUN286" a DOS Extender futtatórendszere. Ez tölti be az előzőleg (-Lp opcióval) védett módban szerkesztett programot, majd végrehajtás után ismét visszaadja a vezérlést a DOS-nak. Így a program a felhasználó számára változatlannak tűnik, míg a DOS az alkalmazás számára védett módúként viselkedik. A Phar Lap 286|DOS Extender 80286-os AT-k mellett futtatható 386, 386SX, 486 és 486SX processzor alapú gépeken is.

A DOS-kiterjesztő nagy előnye, hogy a felhasználó dolgozhat ezután is a megszokott fejlesztőrendszerével (Microsoft C 5.1, 6.0 vagy Borland C++ 2.0) és nyomkövetőjével (CodeView, illetve Turbo Debugger), és nem kell törődnie a programok és az adatok méretével, elhelyezkedésével. Korlátozás nélkül használhatók a BIOS és a DOS összes funkciói, valós, illetve védett módban való végrehajtásukról a kiterjesztő gondoskodik. Továbbra is írhatunk közvetlenül a video-RAM-ba; olvashatunk és írhatunk IN/OUT utasításokkal portokat; használhatunk saját IT rutinokat is.

Ugyanezek a jó tulajdonságok mondhatók el a 386|DOS Extenderről is, de ott további kényelmet ad, hogy a DOS tulajdonképpen 32 bitessé válik, a 64 kbájtos szegmenshatárok eltűnésével 4 Gbájt (4096 Mbájt!) memória címezhető meg lineárisan. A 386|DOS Extenderrel fejlesztett alkalmazások gyorsabbak és kisebbek, használatuk a fejlesztők és felhasználók tapasztalatai alapján átlagosan 20%-os kódmegtakarítást és 50%-os sebességnövekedést eredményezett. Akinél tehát a fenti tényezők jelentős szerepet játszanak, és rendelkezik 386-os vagy 486-os számítógéppel, valamint ilyen kód előállítására alkalmas fordítóprogrammal, annak feltétlenül javallt a 386|DOS Extenderrel való munka. A 386|DOS által a 286-os Extenderrel azonos összegért hozzájuthatunk így egy igazi 386-os fejlesztőrendszerhez is: 386|ASM, 386|LINK, 386|LIB, 386|DEBUG; amely amerikai szakértők (és a szerző véleménye) szerint is sokkal profibb, mint a Microsoft MASM 6.0.

A 386|DOS Extender már nem használható a 286-osnál megszokott eszközökkel (Microsoft C, Borland C++). A saját kódot előállító 386|ASM és 386|LINK mellett elfogadottak még az alábbi, 386-os kódot előállító magas szintű fejlesztőrendszerek:

— Metaware High C és Professional Pascal

— Watcom C8.0/386, Fortran 77/386

— Zortech C++/386

— Silicon Valley Software SVS/C és SVS/Fortran 77

A 386|DOS Extenderrel fejlesztett alkalmazások kompatibilisek a Windows mindhárom módjával, a DESQview 386-tal, a VCPI szabvány alapján működő EMS emulátorokkal (Quarterdeck QEMM-386, Qualitas 386-to-the-

MAX, Microsoft EMM 386, Compaq CEMM), valamint a Windows 3.0 és az MS DOS 5.0 XMS (eXtended Memory Specification) meghajtójával (HIMEM.SYS). Azonban ezek már csak 386, 386SX, 486, 486SX processzor alapú gépeken futtathatók (minden lehetséges — PC AT, PS/2, EISA — buszarchitektúrán).

A 286lDOS Extender, illetve a 386lDOS Extender (és fejlesztőrendszer) ára 49 500 Ft. Minthogy ez a fejlesztő cégek számára már nem jelentős összeg, érdemesebb ezt az egyszeri beruházást megtenni, mint minden projektnél kvalifikált programfejlesztőket felesleges overlay-szervezéssel foglalkoztatni.

A Phar Lap DOS Extender hazai forgalmazója a Gidata Kft.

Az érdeklődőknek e cikk keretein túl is készséggel áll rendelkezésére az írás — és az „IBM PC programozása assembly nyelven" c. könyv — szerzője:

Gidófalvi Zoltán

Érdemjegytőzsde iskolásoknak

Ez húsbavágóan komoly játék — mi több, játszma —, mely a nyereségért folyik hosszú-hosszú hónapokon át. Érdekes vagy szórakoztató? Tanulságos vagy tanulékonnyá tesz inkább? Mindenkinek más a lényeges belőle.

A játék az adott osztály tanulóival folyik, a játékvezető az osztály tanára. A játékot havonta, egy iskolai évben tíz alkalommal érdemes játszani. A játékban a tanulók egyszerre részvénykibocsátók és tőzsdejátékosok. Minden tanuló kétféle részvényt bocsát ki: az ún. átlagrészvényt — amelynek árfolyamát a tanuló átlagának és az osztályátlagnak a viszonya képezi —, és az ún. fejlődésrészvényt, amelynek az árfolyamát a tanuló két tőzsdenap közötti fejlődése/visszaesése határozza meg.

Előkészítés és nyitás

A játék megnyitása előtt a tanárnak a következő adatokat kell a számítógépbe vinnie:

— a játékban részt vevő tanulók névsora

— azon tantárgyak nevei, amelyekben az elért eredmények a játékban számítanak

— a részvények kibocsátási árfolyama (A játék elején, amikor a tanulók még nem kaptak osztályzatot, minden részvény árfolyama azonos. Az árfolyamot a játékot levezető tanár határozza meg.)

— a kötelező részvényvásárlás mennyisége (A játékban részt venni kívánó tanulók számára kötelező egy adott mennyiségben vásárolni saját átlagrészvényeikből. Az ily módon befolyt összeg képezi a bank alaptőkéjét. Ezt a részvénymennyiséget a tanulók a játék lezárásáig nem adhatják el.)

— a minimális árfolyam (A játékban használt számítási módok alapján a részvények árfolyama negatív is lehet. Hogy minden részvény végig játékban maradjon, célszerű egy minimumértéket megadni, amely alá a részvény árfolyama nem csökkenhet.)

— a nyereségadó mértéke (A játékban, mint ahogy a valódi tőzsdén is, az árfolyamnyereséget forrásadó terheli.)

A fenti paraméterek mindegyikét a játékot vezető tanár szabadon határozhatja meg, így a játékot a saját osztályának adottságaihoz igazíthatja.

A nyitás napján a tanulók szabadon vesznek a kötelező mennyiségen felül az átlagrészvényükből, illetve saját elképzeléseiknek megfelelően vásárolnak osztálytársaik fejlődésrészvényeiből.

A tőzsdenapok és a zárás

Egy iskolai évben tíz tőzsdenapot célszerű tartani. Közvetlenül a tőzsdenap előtt a tanár számítógépre viszi a tanulók érdemjegyeit. A szoftver kiszámítja az aktuális árfolyamokat. A tanár feltünteti ezeket az osztályteremben kifüggesztett tőzsdei eredménytáblán. A tőzsdenapokon e tanulók lefolytatják tranzakcióikat, amelyeket a tanár számítógépen rögzít.

Az utolsó tőzsdenap után aktuális a tőzsde zárása, amikor is a bank a napi árfolyamon visszavásárolja a részvényeket. A tanulók a tőzsdenapló segítségével kiszámítják, hogy tranzakcióikkal milyen hozamot értek el.

A motivációs hatás

Az Érdemjegytőzsdének a tanulókra kifejtett motivációs hatásai közül a jelentősebbek:

A játékcsomag részei

— szoftver (egy szoftverrel az iskola összes osztályában folytatható a játék);
— szoftverkézikönyv és játékszabályok;
— tőzsdei eredménytábla (a játékban részt vevő osztályok száma szerint);
— részvények (a játékban részt vevő tanulók száma szerint);
— tőzsdenapló, amelyben az árfolyamok és a tranzakciók regisztrálhatók (a játékban részt vevő tanulók száma szerint).

— A tőzsdenapokon a tanulók megismerik saját és osztálytársaik eredményeit, átlagaikat és az osztályrangsorban elfoglalt helyüket.

— Az Érdemjegytőzsde, akárcsak a valódi, azonnal megmutatja a pozitív és a negatív változásokat, tendenciákat.

— A diákok figyelemmel kísérik egymás pozícióját, minthogy ennek ismeretében játszhatnak eredményesen.

— A játékosok felelősséget éreznek kibocsátott részvényeik tulajdonosaival szemben.

— A tőzsdézők, hogy nagy árfolyamnyereséget realizálhassanak, a rosszabb tanulók részvényeit fogják vásárolni, hiszen itt van lehetőség nagyobb árfolyamemelkedésre. Ez aktivizálja a rosszabb tanulókat. A fejlődésrészvény segítségével a gyenge tanuló a játékban egyenrangú lehet az osztályelsővel: az a gyerek, aki eredményei alapján nem lehet a játék élmezőnyében, ügyesen spekulálva sikerélményhez juthat.

A tanár is érdekelt, hiszen sokat megtud az árfolyam-alakulásokból és a tranzakciókból. A szoftver rendelkezik egy „Információk" menüponttal is, amelyben a tanár az összes lehetséges kimutatást és statisztikát megkaphatja, amely a gépbe bevitt adatokból származtatható.

Sal István

Megtépázva a dolgozatíratás réme

A program a hozzá kapható feladatgyűjteményekből az osztály minden tanulójának egyedi dolgozatot állít össze. A feladatok lehetnek: különböző feladatok egzakt rész- és végeredménnyel; azonos feladat generált változóval; feleletválasztás; relációanalízis; igaz-hamis állítás; kifejtő kérdés.

A teszt jellegű feladatoknál egy kérdésre több feladat is irányulhat, az értékelésnél a program a megoldásokat együtt kezeli.

A feladatok összeállításánál megadható: a tananyag; a feladatok nehézségi foka; a részeredmény figyelembe vétele; a feladatok pontszáma; osztályozás az elért pontszámok alapján. A feladatok összeállítása után a tanár névre szólóan kinyomtathatja a dolgozatlapokat — minden tanulónak „testre szabottan".

A tanárnak lehetősége van arra, hogy ne a szoftverhez készült feladatgyűjtemény feladatait vagy azokon felül más kérdéseket tűzzön ki. A programba bevihet saját feladatokat, sőt saját kollekciót is előállíthat.

A program nagyban egyszerűsíti a dolgozatokkal kapcsolatos fáradságos munkát. Tetszőleges számú osztály dolgozatait képes összeállítani, feldolgozni és tárolni. Az értékeléshez a programban lehívhatóak a dolgozatok, és a diák részeredményeit beírva a program elvégzi a teljesítmény értékelését. Az értékelésnél megadhatók bizonyos tűréshatárok, illetve ha a tanár úgy kívánja, saját maga adhat minősítést. Kifejtő kérdéseknél eleve a tanár osztályoz.

A diákok a dolgozatokat névre szólóan kapják, az adatok, a kérdések vagy a témák nyomtatva szerepelnek a lapokon — ez kiküszöböli a másolásból és diktálásból eredő hibákat. A tanulók nem kapnak azonos feladatot, így nincs lehetőségük a másolásra.

A fentiekből kitűnik, hogy egy iskolában minden tanár használhatja a programot — akár egyetlen számítógépen is. (A játék és a tanári szofver kifejlesztője és forgalmazója a Logo-Press BT.)

INFORMÁCIÓKÉRÉS: 25 ▶

Suli-bank a tanáriban

Az iskolai adminisztráció is része annak a területnek, mely az oktatási intézmények szférájába tartozik. A iskolavezetésnek minden évben többször is visszatérő gondja a statisztika. Segítsünk ebben a dologban!

Magam is találkoztam olyan programokkal, amelyek különböző statisztikai, adminisztrációs munkákat hivatottak megkönnyíteni. Hátrányként éreztem ezeknél a programoknál, hogy nem épülnek be a oktatási intézmények mindennapjaiba — csak szolid, kockázatmentes területeket vállalnak fel. (Ha valami nem egyezik, a tisztelt felhasználó követett el hibát.)

Közel két évi fejlesztőmunka eredményeként elkészült egy „SULIBANK". Ez a statisztikai funkciókon túl a térítéses szolgáltatások (napközi, menza stb.) időnként kaotikus nyilvántartásait is magában foglalja. A kétévi fejlesztés fő oka, hogy a gyakorlat néha olyan meglepő fordulatokat prezentált, amilyeneket csak változó és infláló világunk képes (például: visszamenőleges áremelés, hóközi áremelés, hogy ne is a legcifrábbakat említsük). A rendszer menzát, ebédet rendel, felszólítást javasol, havi pénzforgalmi zárólistát készít (a tényleges befizetések, hátralékok és túlfizetések figyelembevételével), és még sorolhatnám az osztály-, napközicsoport- és egyéb listákat.

Mivel az iskolákban nem számítástechnikában jártas hölgyek vagy urak intézik ezeket az ügyeket, ezért a programot például kétféleképpen is használhatják. Vagy a meglévő ügyviteli gyakorlathoz illeszkedve, vagy a programrendszer úgynevezett „gépi" ügyvitelének felhasználásával. Összesen 47 menüpont szolgálja a kényelmesebb iskolai adminisztrációt. Adatbázisai nyitottak, bármely adatbázis-kezelővel hozzáférhetők. Saját „nyomozó" funkcióval rendelkezik, hogy egyeztesse adattárait, ha esetleg összekuszálták azokat az adatbázis-kezelővel. (A részletesebb leírás a mágneslemezen Sulibank néven megtalálható.)

Schneider Ferenc

A lélek tükre — széttörve

A PC-k felboncolásának első két üteme után, jelen alkalommal a képernyő kerül kés alá. A forgalomban lévő displaykártyák választéka nagyon széles. Itt a minden esetben érvényes megállapításokon kívül részletesen csak a CGA- és a monokróm kártyáról lesz szó. A többi vezérlőtípusnál meg kell elégedni a BIOS-on keresztül elérhető funkciók ismertetésével. (Azok az ábrák és táblázatok, amelyekre a szöveg több helyen is utal, a mágneslemezen kaptak helyet. Az írás áttanulmányozása előtt ajánlatos ezeket kinyomtatni.)

Az IBM PC alapkiépítésben nem tartalmaz semmilyen képernyővezérlő áramkört. Ha a munkánk eredményét látni is akarjuk, kénytelenek vagyunk vásárolni egyet, és ezt a bővítőhelyek egyikébe kell behelyezni. A megfelelő típusú (a választott vezérlőkártyához való) monitor csatlakoztatása után már nézhetjük is a képet. Az elterjedtebb vezérlő- és monitortípusokat, valamint azok jellemzőit az 1. táblázat tartalmazza.

Nem szerepel a táblázatban az EGA-mód azért, mert — véleményem szerint — nem népszerű igazán, és a Hercules-mód azért, mert az IBM (elterjedtsége ellenére) nem ismeri el szabványosnak. Egyébkényt a Hercules vezérlők és képernyők text-működésmódja teljesen megegyezik a monokróm rendszerekével, a megfelelő DOS- és BIOS-funkciók használatával dolgozhatunk a képernyőn. A Hercules grafikus módot azonban sem a DOS, sem a BIOS nem támogatja; ha használatához ragaszkodunk, akkor saját rutinokkal kell megoldani a kezelését.

A „retina"

Az IBM PC-kben a képernyő tartalmának tárolására egy kijelölt memóriaterület szolgál: ezt nevezik képernyő- vagy video-RAM-nak. A video-RAM monokróm adapter esetén a B0000 fizikai címen kezdődik, és minden karakterpozícióra 2 bájtot — összesen (80x25x2) 4000 bájtot — foglal le. Az ilyen típusú vezérlő csupán egy megjelenítési módban tud dolgozni. A képernyő textmódban működik, 80 oszlopot és 25 sort jelenít meg egy időben. A video-RAM és a képernyőpozíciók közti összerendelés sorfolytonos. Az első 160 bájt az első, a második 160 bájt a második sorokat jelenti, és így tovább. Az adott soron belül az első két bájt az első, a második két bájt a második stb. karaktert határozza meg. Az egy karaktert meghatározó két bájt közül az első a karakter ASCII-kódját tartalmazza. A második bájt az attribútum kódja. E bájt alsó négy bitje a tinta színét, felső négy bitje pedig a hátteret adja (lásd a 3. táblázatban).

CGA-vezérlő esetén a video-RAM a B8000 fizikai címen kezdődik. Ha a képernyőt 80 oszlopos, 25 soros üzemmódban használjuk, akkor az egyes bájtok és képernyőpozíciók összerendelése megegyezik a monokróm adapternél leírtakkal.

A különbség csupán az attribútumbájt színkódjainak értelmezésében van (3. táblázat).

A CGA 40x25 karakteres szöveges módban a video-RAM beosztása nagyon hasonló a 80x25 módéhoz. A különbség csak annyi, hogy egy sor tárolására csupán 80 bájtot kell felhasználni, és így egy teljes képernyő 2000 bájt memóriát igényel.

A CGA-vezérlőt a két szöveges mód mellett grafikus megjelenítésre is használhatjuk. Ebben az esetben a video-RAM két részre bomlik. A terület első felén a páratlan sorokat, a másik felén pedig a páros sorokat tárolja a rendszer. Az egyes sorokat reprezentáló területek normál sorrendben követik egymást. Az első sort leíró memóriaterület után a harmadik, majd az ötödik... stb. következik. A százkilencvenkilencedik sor után a második, negyedik, hatodik... stb. van tárolva. A grafikus módokban minden egyes sort 80 bájt ír le.

640x200-as felbontás esetén az egy sort leíró 80 bájt minden bitje egy képernyőpontnak felel meg. Ha a bit 1 értékű, akkor a pont világít, egyébként háttérszínű. Az első bájt 7. bitje jelenti a sor első pontját, a 6. bit a sor második pontját, a második bájt 7. bitje a sor 8. pontját, és így tovább. 320x200-as felbontású módban minden egyes képpontot két bit határoz meg. A képernyőpontok és a memóriaterületek egymáshoz rendelése megegyezik a nagyfelbontású módnál leírtakkal. Tehát az első bájt 7. és 6. bitje az első sor első pontját, az 5. és 4. bit a második képpontot, a második bájt 7. és 6. bitje az ötödik képpontot határozza meg. Az egyes képernyőpontok így négy különböző színt vehetnek fel. Ha a képernyőpont színkódja 0, akkor a pont háttérszínű lesz (lásd a color select regiszter leírásánál), ha 1, 2 vagy 3, akkor az érvényes paletta (lásd a color select regiszter leírásánál) 1., 2., illetve 3. színét veszi fel. (A memória és a képernyő összerendelését szemlélteti a 2. ábra.)

Az IBM PC az A0000 és C0000 fizikai címtartományba eső memóriát fenntartja a video-RAM részére. A fent említett 128 kbájt méretű memória a képernyővezérlő kártyán helyezkedik el. Nem mindegyik kártyára építik fel a 128 kbájt RAM-ot. A monokróm kártyán 4, A CGA-n 16 kbájt kapott helyet. Mivel egyes megjelenítési módok ennek csak a töredékét használják fel, itt lehetőség van több képernyőnyi információ tárolására. Ezt a lehetőséget a BIOS is támogatja. A megvalósítás módjára a 6845-ös IC R12 és R13 regiszterének leírása, valamint a megfelelő BIOS-funkciók mutatnak rá.

A „szemizmok"

A monokróm, Hercules- és CGA-kártyákban a képernyő fizikai vezérlését egy Motorola 6845 típusjelű integrált áramkör végzi. Ez az IC 18 db írható belső regiszterrel rendelkezik. A belső regiszterekbe írt értékek határozzák meg a képernyőnek küldött jelek tulajdonságait. Ezek a regiszterek a következő módon érhetőek el: először az address regiszterbe (i/o címe — MONO: hexa 3B4 — CGA: hexa 3D4) kell

írni az elérni kívánt belső regiszter sorszámát, ezután az adatregiszteren (i/o címe — mono: hexa 3B5 — CGA: hexa 3D5) keresztül elérhető a regiszter. Ez utóbbi címen a manipulációk mindig az utoljára megadott sorszámú belső regiszterre vonatkoznak.

A 6845-ös belső regisztereinek ismertetése előtt tisztázni kell néhány fogalmat. A képernyőre a jeleket egy elektronsugár rajzolja ki. Ez a sugár a bal felső sarokból elindulva soronként pásztázza végig a képernyőt. Amikor egy sor végére ér, akkor a vízszintes szinkronjel visszavezérli a következő sor elejére. Ha a legalsó sor megjelenítésével is végzett, akkor a függőleges szinkronjel hatására visszatér a bal felső sarokba, majd kezdi elölről a képfrissítést. A két függőleges szinkronjel közt eltelt idő a képfrekvencia, a két elemi sor frissítésének megkezdése — két vízszintes szinkronjel — közt eltelt idő pedig a sorfrekvencia. (Ha arra szánjuk magunkat, hogy az alább ismertetett jellemzők gyári beállítását megváltoztassuk, akkor alaposan tisztában kell lennünk a vezérlőkártya és a monitor képességeivel. Rosszul beállított értékek maradandó károsodást okozhatnak, sőt végzetesen tönkre is tehetik a monitort.)

A belső regiszterek általában nyolcbitesek és csak írhatóak — az ettől való eltéréseket külön jelezzük.

R0 — Két elemi sor frissítésének megkezdése közötti idő. A sorfrekvencia reciproka. Mértékegység az egy karakterszélesség kirajzolásának ideje. A kívánt értéknél eggyel kevesebbet kell írni.

R1 — Az egy sorba rajzolandó karakterek száma.

R2 — A sorfrissítés kezdete és a sorszinkronjel kezdete közötti idő. Mértékegység: mint R0-nál.

R3 — A vízszintes szinkronjel hossza. Mértékegység: mint R0-nál. A regiszter csak négybites.

R4 és R5 — Ez a két regiszter határozza meg a képfrekvencia nagyságát. Az értékeket a következő módon határozhatjuk meg. Az (összes elemi sorok száma)/(R9+1) hányadosa kerül R4-be, a maradéka R5-be. Az „összes elemi sorok" itt nemcsak a kijelzett sorokat jelenti, hanem a két függőleges szinkronjel között kitehető sorok számát. R4 hét-, R5 pedig ötbites.

R6 — A kijelzésre kerülő karaktersorok száma. Nem feltétlenül egyenlő R4-gyel. Kisebb lehet, csak nagyobb nem. A regiszter hétbites.

R7 — A képernyő frissítésének kezdete és a függőleges szinkronjel kezdete közti idő. Mértékegység az egy karaktersor megjelenítéséhez szükséges idő ((R9+1)*R0). A regiszter hétbites.

R8 — Kétbites regiszter. Szokásos értéke 2. Más értékek beállításával a képfrissítés stratégiáját változtathatjuk meg.

R9 — Az egy karaktersort alkotó elemi sorok száma mínusz 1. A regiszter ötbites.

R10 — Hétbites. Alsó öt bitje adja meg, hogy a kurzor a karakter hányadik elemi sorában kezdődjön. Az ötödik bit a kurzor láthatóságát vezérli, a hatodik nincs bekötve.

A két felső bit eredetileg egész mást jelent, itt a speciális IBM PC-s megvalósítás szerepel.

R11 — Azt adja meg, hogy a kurzor a karakter hányadik elemi sorában végződjön. A regiszter ötbites.

R12 és R13 — A képernyőterület kezdetére mutató pointer. IBM PC-s megvalósításban relatív a képernyőterület kezdetéhez képest. R12 hatbites, a cím magasabb helyi értékű összetevője. R13 nyolcbites. A címzés egysége nem bájt, hanem szó. Így nem tudjuk a karakterkódok és attribútumok jelentését megcserélni. E regiszterekkel végzett ügyes manipulációval egy nagyon gyors hardverscrollozást lehet megvalósítani.

R14 és R15 — Ugyanolyan szervezésű mutató, mint R12 és R13. R14 hatbites, R15 pedig nyolc. A kurzor helyét mutatják. A regiszterek írhatóak és olvashatóak.

R16 és R17 — A fényceruza kezelésében játszanak szerepet. E két regiszter tartalmazza a fényceruza által címzett pozíció relatív címét. A regiszterek csak olvashatóak.

A CGA-adaptert még három porton keresztül programozhatjuk. A hexa 3D8 i/o címen elérhető módválasztásbájt csak írható. A 3D9 címen lévő színválasztó bájt csak írható. A 3BA címen lévő státusbájt csak olvasható. Az egyes bitek jelentését a 4. ábra mutatja. monokróm adapter esetén a módválasztásbájt a 3B8 címen, a státuszbájt pedig a 3BA címen helyezkedik el. A bitek kiosztását a 4. ábra mutatja.

A képernyők vezérlésének bemutatásából már csak a BIOS-on keresztül elérhető funkciók leírása van hátra.

A BIOS képernyőkezelő funkciói a hexa 10 sorszámú megszakításon keresztül érhetőek el. A megszakítás húsz dokumentált funkcióval rendelkezik. A megszakítás aktivizálásakor a végrehajtatni kívánt funkció sorszámát az AH regiszterben kell elhelyezni. A többi regiszterbe írandó értéket az 5. táblázat mutatja. A megszakítások sorszáma és funkcióik:

0 — A képernyő üzemmódjának meghatározása. A funkció törli, majd az AL regiszterbe helyezett érték szerinti üzemmódba állítja a képernyőt. Nem szolgáltat visszatérési értéket.

1 — A képernyőkurzor megjelenési formája (kezdő és végsora) állítható be. Ha a kezdősor nagyobb a végsornál, akkor a kurzor lyukas lesz. Ha mindkét koordináta nagyobb, mint a legalsó sor száma, akkor a kurzor nem látható. A funkció a 6845-ös R10 és R11 regisztereit írja. Nem ad visszatérési értéket.

2 — A képernyőkurzor pozicionálása. A DOS minden lehetséges képernyőlap kurzorpozícióját nyilvántartja, és a kijelzett lap kurzorkoordinátáit írja be a 6845-ös R14 és R15 regisztereibe. Ezzel a funkcióval nem csak a kijelzett lap kurzorkoordinátái állíthatók be. Nincs visszatérési érték.

3 — Adott képernyőoldal kurzorjellemzőinek a lekérdezése. A funkció megadja a kurzor kezdő és végsorát, valamint a koordinátáit. A kezdő és végsor minden képernyőlapon azonos lesz, de a koordináták külön-külön vannak nyilvántartva laponként.

4 — Fényceruza-pozíció lekérdezése. A funkció megmondja, hogy a fényceruza aktív-e, és ha igen, akkor melyik pontra mutat. A visszaadott koordináták az adott üzemmód felbontása szerinti számértékeket tartalmazzák. Pontatlanságuk kb. plusz-mínusz 4%.

5 — A kijelzendő képernyőoldal kiválasztása. A funkció a kurzorpozíciót is beállítja az adott képernyőlapnál nyilvántartott helyre. Nem változtatja meg az egyes képernyőlapok tartalmát.

6 — Szövegsorok felfelé görgetése. A funkció egy négy sarokpontjával meghatározott képernyőablakot görget. A gördítendő sorok számát meg kell adni. Minden sornyi görgetésnél a legfelső sor elveszik, az alsó, belépő sorban pedig szóközkarakterek vannak. Ha a gördítendő sorok számát nullának választjuk, akkor az ablak szóközökkel töltődik fel (törlődik). Ha a gördítendő sorok számát nagyobbra választjuk, mint az ablak magassága, akkor a BIOS verziójától függően különböző katasztrofális dolgok történnek.

7 — Ugyanaz, mint az előző funkció, de az ablakot lefelé görgeti.

8 — Adott képernyőpozíción lévő karakternek és attribútumának a meghatározása. Nemcsak az aktív, hanem bármely képernyőlapon képes dolgozni. Grafikus üzemmódokban is hívható.

9 — Karakter és attribútum kiírása adott képernyőlap adott pozíciójára.

Grafikus üzemmódban az attribútum helyére a karakter színét kell megadni. 640x200-as üzemmódban 0 vagy 1, 320x200-as üzemmódban 0, 1, 2 vagy 3 lehet. A karakter kiírását a megadott számszor hajtja végre. Grafikus üzemmódban a kiírt karaktereknek el kell férniük egy sorban. A különböző vezérlőkódok is ASCII-karakterekként lesznek kiírva. A kurzor pozícióját nem aktualizálja.

A — Ugyanaz, mint az előző funkció, csak az attribútumot nem adhatjuk meg. Az eredi beállítás marad érvényben.

B — Háttérszín (a színválasztó bájt 0., 1., 2. bitje) vagy az érvényes paletta (a színválasztó bájt 5. bitje) beállítása. Bővebben lásd az adott bitek leírásánál.

C — Képernyőpont kirajzolása grafikus üzemmódban. A megadott képernyőpontot a megadott színűre állítja. Ha a megadott szín legmagasabb bitje 1, akkor a kirajzolandó és a már kint lévő pont között kizáró vagy (xor) műveletet hajt végre. Ebben az esetben biztos, hogy változni fog valami az adott ponton.

D — Képernyőpont kiolvasása grafikus üzemmódban. A grafikus képernyő egy adott pontjának a színét adja meg.

E — Karakter kiírása az aktuális kurzorpozícióra. Szöveges üzemmódban meghatározhatjuk a képernyőlap számát, az attribútum pedig megegyezik az előző karakterével. Grafikus üzemmódban megadható a karakter színe is. A kiírás után a kurzor pozíciója aktualizálódik. A vezérlőkaraktereket vezérlési funkciójuknak megfelelően értelmezi és végrehajtja.

F — Az aktuális üzemmód lekérdezése.

10 — (csak IBM AT-nél!) Karakterlánc kiírása a megadott oldal megadott pozíciójától kezdődően. A kiírandó karakterfüzért egy pufferben kell elhelyezni, és a puffer címét kell átadni a funkciónak. Ha az attribútumbájtokat is a pufferbe írjuk, akkor a páratlan sorszámú bájtok az ASCII-kódok, a párosak pedig az attribútumok.

Fridl György

3M HUNGÁRIA KFT. XLIX

A **3M** Hungária Kft. a 3M mágneses adathordozók teljes választékát kínálja partnereinek:

HARDVER | SZOFTVER | IRODATECHNIKA | EGYÉB

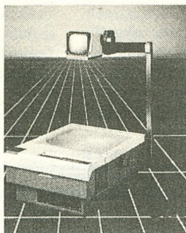


Mágneses adathordozók teljes választéka:

- diskettek,
- streamerkazetták,
- TK 50, TK 52 kazetták,
- IBM 3480 kazetták,
- komputerszalagok,
- optikai diskek,
- kiegészítők.

Irodatechnikai és irodai eszközeink a legmagasabb igényeket is kielégítik.

- LCD folyadékkristályos kijelzők,
- írásvetítők,
- fóliák, tollak,
- Post-it™ öntapadós jegyzettömbök.



3M Hungária Kft.
1054 Budapest,
Vécsey u. 4.
Tel: 111-7860
Telex: 22-3234 trlem h
Fax: 163-3220

Az innováció Önnek dolgozik™

INFORMÁCIÓKÉRÉS: 19 ▲



KELLEMES KARÁCSONYI ÜNNEPEKET KÍVÁNUNK, ÉS NE FELEDJÉK EL A '92-ES ÉVBEN SEM, HOGY AZ ÁR AKKOR JÓ, HA ÖSSZHANGBAN VAN A MINŐSÉGGEL.

Barex computer kft.

BAREX COMPUTER KFT.
1054 Bp., Bajcsy-Zsilinszky út 54.
Tel.: 131-0946 Tel./Fax: 111-6025

INFORMÁCIÓKÉRÉS: 21 ▲

szoftver ABC

☎ : 201-6891, 201-2011/131
Fax : 201-8619
✉ : 1277 Budapest 23. Pf: 45.

A legjobbat a legjobbaknak, azonnal!

Raktárról szállított szoftvereink:

Szoftver:	**Ár (+ÁFA)**
Clipper 5.01	68 990,–
Corel Draw 2.0	50 900,–
Foxpro 2.0	61 490,–
Foxpro Lan	91 000,–
Foxpro Runtime	54 400,–
Framework IV	64 000,–
Harvard Graphics	48 000,–
Laplink V. 3.	12 900,–
MS-C 6.0	40 900,–
MS-DOS 5.0	8 400,–
MS-Excel 3.0	40 900,–
MS-MASM 6.0	15 200,–
MS-WORD for Xenix and Unix	99 900,–
MS-Windows 3.0	10 900,–
MS-Windows Device Dev. Sys.	35 900,–
MS-Windows (Szoftver) Dev. Sys.	35 900,–
MS-Word 5.5	33 110,–
MS-Word for Windows	41 490,–
Micrografx Charisma	37 900,–
Micrografx Designer 3.1	44 900,–
Nantucket Tools II	54 900,–
Norton Commander 3.0	12 000,–
Norton Utilities 6.0	13 000,–
Novell 2.2 10 user	130 000,–
Novell 2.2 5 user	58 900,–
Novell 386 3.11 100 user	475 000,–
Novell 386 3.11 20 user	230 000,–
PC Tools 7.0	12 000,–
Paradox 3.5	38 900,–
Printer Assist	16 900,–
Procom Plus	13 500,–
Quattro Pro 3.3	16 000,–
SCO Foxbase for Xenix	103 900,–
SCO Foxbase for Xenix Runtime	34 900,–
SCO TCP/IP for Unix 386	24 900,–
SCO Unix 3.2 Dev. Pack	78 900,–
SCO Unix 3.2 Op. Sys.	55 600,–
SCO Xenix 386 Op. System	70 600,–
SCO Xenix 386 Text Pr. Pack	24 900,–
Superbase 4 For Windows	50 900,–
Turbo Pascal for Windows	22 900,–
Ventura Publiser Gold for Windows	74 900,–
WordSTAR 6.0	36 900,–
Wordperfect 5.1	43 900,–

Még sok száz SZOFTVER rövid szállítási határidővel!

Kérje részletes katalógusunkat!

Érdeklődjön rendszeresen, a Szoftver ABC-ben mindig van akció!

INFORMÁCIÓKÉRÉS: 13 ▲

Hozzáférés — két akadályon áthágva III.

Az akadályok kiegészítése

Az előző részben végül is csak felsoroltuk az első szűrési lépcsőben megoldandó feladatokat. Ezek képezik azokat az akadályokat, melyek megnehezíteni hivatottak az illetéktelen hozzáférést a rendszerünkhöz. Most további nehézségeket támasztunk.

Online kódolás

A célunk tehát az, hogy a merevlemezünkön található információt rendkívüli módon titkosítsuk, lekódoljuk. Alapállapotban a lemezen lévő információ állandóan olvashatatlan. Ahhoz viszont, hogy lemezről olvasáskor a memóriát ne etessük meg mindenféle zöldséggel, vissza kell állítanunk az eredeti állapotot, azaz vissza kell kódolnunk a beolvasott adatokat. Ugyanez a helyzet, amikor a lemezre kívánunk írni, ilyenkor ugyanis a kiírandó információt kell lekódolnunk, így amikor már a lemezre kerül, értelmezhetetlen lesz, akár az egész lemez. A következő dolgokat kell megoldanunk.

1. Rögtön a rendszer installálásakor: egy program elvégzi a teljes lemezegység kódolását egy előre megalkotott, a programhoz csatolt kódolótábla segítségével.

2. A rendszer élete során folyamatosan: beláncolódik egy program a 13h-s megszakításra, ahol az előzőekben leírtakat kell elvégeznünk:

— ha valaki merevlemezre akar írni, akkor lekódoljuk a kiírandó adathalmazt;

— elugrunk az eredeti 13h-s rutinra;

— ha merevlemezről akar olvasni, akkor visszafejtjük a kiírandó adathalmazt.

Az 1. ponthoz csak annyit, hogy ez a kódolótábla tárolva van ugyan, de maga is a felismerhetetlenségig kódolva. Minden egyes rendszerhez, és így minden merevlemezhez más kódolótábla tartozik, tehát ha megfejtenek egyet, akkor a visszafejtés elve megvan ugyan, de akkor a másik egységen lévő információ hozzájutásához meg kell tudni az ott létező, a másiktól eltérő kódolótáblát. Ha kártyánk van, akkor ezt a táblát ott célszerű elhelyezni, egyébként a merevlemezegység egy kevéssé átjárt helyére kerül, ahol még a madár sem jár, de az író-olvasó fej igen. Akkor sem lenne baj, ha nem lenne kódolva a tábla, hiszen az algoritmus nem ismert, mely nélkül pedig csak a játszadozás marad hátra, vagyis annak a megfejtése, hogy annak a néhány számnak mi köze van a lemezen tárolt adatokhoz.

A 2. pont az érdekes számunkra, hiszen számtalan problémát vet fel. Itt van mindjárt az, hogy mikor láncoljuk be magunkat. Mindenképpen az lenne az igazi, hogy legelsőnek, vagyis még mielőtt bárki hozzáférhetne a géphez, hiszen a beláncolódásunk előtt a gép olyan, mint az éjszakára nyitva hagyott ékszerbolt. Az őrségnek a biztonságot nyújtó zárt ajtónak a legelső látogató megérkezése előtt már posztolnia kell. Ha kártyát is alkalmazunk, akkor ez nem probléma, hiszen az ott elhelyezett rutin előbb indul el, mint az A meghajtó kezdőszektorán elhelyezett program. Költséges mivolta miatt azonban ez nem mindig oldható meg, így egy trükkhöz kell folyamodnunk. A kódoló programocskánkat az operációs rendszer betöltése közben becsempészhetjük a tárba. Ez előtt tehát csak a floppyról történő indítás van, de a merevlemezzel akkor sem tud senki mit kezdeni. Olyan ez, az előző példából kiindulva, mintha bejutott volna ugyan valaki az ékszerboltba, de vagy nem tudja, hogy ott óriási értékek vannak, vagy ha tudja, akkor nem találja meg azokat, mert mondjuk éppen nyalókának, gombostűnek vagy esetleg porszívónak vannak álcázva. Ehhez arra van szükség, hogy már a betöltő szektor is eltérő legyen a megszokottól, tehát alkalmas legyen a mi rutinjaink betöltésére.

Egy másik probléma: a beláncolódásnak az a feltétele, hogy a programunk bent maradjon a memóriában. Ennek látszólag semmi akadálya, hiszen elvileg a memória megtartja tartalmát, feltéve persze, ha rendszeresen, mondjuk kétezred másodpercenként kap frissítőt. A nehézség abban áll, hogy a DOS mit sem törődik azokkal a programokkal, amelyek a tudomásán kívül garázdálkodnak a memória lankás tájain, és gátlástalanul rátölthet egy másik, talán számára kevésbé kedvelt programot. Azért nem tudja, hogy mi bent vagyunk a memóriában, mert akkor még DOS sincs, amikor mi már létezni akarunk. Ha kártyánk van, akkor az ott elhelyezett ROM-on vagy RAM-on keríthetünk neki helyet; de mi van akkor, ha nincs?

Kártyacsata

Ekkor hazardírozunk egy kicsit. Megpróbálhatjuk a grafikus kártyán fellelhető szabad területeket elérni, hiszen ezekben a pillanatokban szöveges üzemmódban dolgozunk. Ez körülményes, hiszen többféle grafikus kártya létezik, és a rendszer veszítene kompatibilitásából, ha külön változatokat kellene készíteni minden egyes kártyára. Lekérdezve a kártyát persze megoldható, hogy minden típusú kártyára megírjuk ezt a rutint.

A másik megoldás az, hogy a memória közepére beírjuk magunkat, és drukkolunk, hogy a DOS nehogy oda töltődjön. Ennek kicsi az esélye, hiszen az először betöltődő DOS-állományok, amelyeket a bootszektor hoz be, a memória legaljára kerülnek, ahol már nincs rendszerleíró adat. (Ilyen adat például a megszakítástábla és a BIOS-változók, no meg a BASIC rendszer rendelkezésére álló egy-két bájt.)

Mindkét megoldásnál a DOS rendszer létezésénél újra be kell láncolódnunk, hogy most már ő is tudomást szerezzen rólunk. Azért kell a DOS rendszer előtt már beláncolódnunk, hiszen akkor a DOS-rendszerfájlok és a COMMAND.COM is kódolatlanul heverne a lemezen, ami egy sor újabb problémát vetne fel. Ha kártyánk van, akkor ez a második beülés felesleges, hiszen a ROM-ba igen nehezen tudna programot tölteni a DOS...

Egyéb kérések megszűrése

Bizonyos kérések ugyancsak a BIOS-tól várják a megváltást, de nem biztos, hogy ezek teljesítése nekünk is megváltást hozna...

Az előbbiekben csak megszállottan kódoltunk, de azt még nem néztük meg, hogy a gépnél ülő barátunk egyáltalán írhat-e a lemezre, akár a hajlékonyra, akár a fixre. Márpedig egyikőnknek sem kellemes, ha munkánk határidejének előestéjén valaki kapirgál a FAT-ben vagy a bootszektorban. De nem csak a lemezre írást és olvasást kezelhetjük ebben a megszakításban, hiszen ha jól körülnézünk, az előbbiekben felsorolt alfunkciókból jócskán csemegézhetünk. Itt van rögtön például a formattálási probléma. A különböző esetekre másképpen ajánlatos reagálnunk.

— Ha írni vagy olvasni akar az idegen a merevlemezen, akkor (mielőtt a kódolórutinra engednénk rá a vezérlést) meg kell vizsgálnunk, hogy nem a merevlemez valamely bootszektorát akarja-e netalántán írni vagy olvasni, vagy nem éppen a partíciótáblához akar-e nyúlni. Ha valaki mégis rászánná magát, hogy olvassa ezeket a zónákat, akkor egy verset kap vissza az olvasandó adatok helyébe.

— Ha írni vagy olvasni akar a hajlékonylemezen, akkor ezt csak akkor engedjük meg, ha a rendszergazda ezt jóváhagyja; tehát, ha szabad neki, akkor kódolás nélkül továbbadjuk az eredeti 13h-snak ezt a kérést.

— Ha formattálni akar, ugyanez a helyzet: vagyis ha a rendszergazda biztosította neki ezt a jogot, akkor minden további magyarázkodás nélkül tolmácsoljuk ezen kérését az eredeti 13h-snak. Természetesen a merevlemez formattálását semmiféleképpen sem engedélyezzük.

Itt is fontos, hogy mikor láncolódunk be. Ha kártyánk van, akkor ezt is célszerű a kártyára rátenni, egybefűzve az online kódolást megvalósító rutinnal. Ha nincs kártyánk, akkor ezt elég egyszer, a DOS rendszer betöltése után beláncolni a 13h-s megszakításba.

Kedvenc printerünk levédése

Teljesen hasonlóan kell kezelni ezt a jogosultságot, mint a 13h-s megszakításon átfutó kérések megszűrését. Rá kell tehát ülni a 17h-s megszakításra, és figyelni, jön-e hozzánk printerhasználati igény. Csak akkor engedjük tovább az eredeti rutinnak a kérést, ha a gépnél ülő felhasználó megkapta a jogosultságot. A beüléssel teljesen azonos problémák merülnek fel, mint a 13h-s esetében — célszerű tehát azzal együtt beültetni a tárba. Viszont itt nem lényeges, hogy a DOS létezése előtt már bent legyen a tárban, elég, ha a COMMAND.COM-mal egyetemben lesz elindítva.

A rendszerbetöltés letiltása az A meghajtóról

Abban az esetben lehetséges ez, ha kártyával rendelkezünk. A kártyánkon elhelyezett rutinok az előzőekben leírt módon már az A meghajtó vizsgálata előtt beülnek a memóriába. Ilyenkor tehát csak a BIOS-t kezelő rutinjainkat helyezhetjük el a kártyán, mivel a DOS még nem létezik, ezért megszakításaiba sem tudunk beláncolódni. A printert levédő rutinunkat az előzőekkel ellentétben nem a 13h-sba, hanem a 17h-s megszakításra láncoljuk be. A következő rutinnak is a kártyán a helye. Ez nem csinál mást, mint kinéz az A meghajtóra, és megvizsgálja, hogy van-e ott olyan jel, amelyet mi helyeztünk el. Ha nem létezik ilyen, akkor a továbbiakban a 13h-s megszakításban nem engedélyezzük az A meghajtó használatát, tehát maga a BIOS betöltő rutin sem tud semmit betölteni. A rendszer betöltésével természetesen a felhasználó jogosultságának megfelelően fogjuk engedélyezni vagy éppen letiltani a lemezegység használatát. Ha viszont a jelszócska létezik, simán továbbadjuk az eredeti rutinnak a kérést az A meghajtóval kapcsolatban.

Meglepetés a vírusok ellen

Nem azért neveztük meglepetésnek, mert nekünk jutott volna eszünkbe először, és hogy ez olyan csodaszer, amely megszüntetné a vírusfertőzés veszélyeit. Nem másról van itt szó, mint a megszakítástáblázat folytonos figyeléséről: ha abban valami változás történik, vagy szólunk a felhasználónak, hogy vigyázzon (mert talántán ez a tudta nélkül történik), vagy egyszerűen visszaírjuk az eredeti vektorcímet — tehát teljesen megóvjuk a felhasználót az ilyen titkos tevékenységektől.

A vírusok általában úgy tudnak észrevétlenek maradni, és bűnös akciókat eszközölni, hogy titkosan beláncolódnak valamely megszakításba, és — miközben esetleg a felhasználó betölt egy programot — elkezdenek terjedni. Ezt a gépnél ülő nem veszi észre, mivel látja ugyan, hogy mozog az író-olvasó fej, de azt hiszi, hogy saját programjával foglalkozik a DOS, közben pedig a vírus elragadta tőle a vezérlést. Tehát a felhasználó megmondhatja, hogy most olyan feladatot fog végezni, amelynél fontos, hogy ne zavarják meg különböző háttértevékenységekkel. Ilyenkor mi nem engedjük, hogy átírják a megszakítástáblát. Olyan üzemmód is választható, hogy átírhatók ugyan a megszakítások, de szólunk, hogy vigyázat, mert valami furcsa dolog történt.

A megfigyelést egy speciális megszakítás átírásával végezzük el, az 1Ch-val. Erről már esett szó a BIOS-megszakítások felsorolásánál. Ez a felhasználó által nagyszerűen használható timermegszakítás, mely minden 1/18 másodpercben meghívódik. Tehát mi minden ilyen ütemben ellenőrizzük a megszakítástábla egy kis részét. Azért egy kis részét, mert itt nem szabad sok időt eltöltenünk, mivel az jelentősen lassítaná a gép munkáját.

Van egy abszolút határ is, amelynek a túllépése a gép lefagyását eredményezi. Ez azért van, mert ez a megszakítás a 8h-s hardvermegszakításból aktivizálódik, tehát az mindig végrehajtódik — függetlenül a mikroprocesszor munkájától.

A megszakításnak általában akkor van vége, ha egy IRET gépi kódú utasítás feldolgozásához lát a központi egység. De nem így van ez az 1Ch-nál abban az esetben, ha ott túl sok időt vesztegetünk! Mivel a megszakításvezérlő szól egy újabb 1/18 másodperc elteltével a mikroprocesszornak — hogy most pedig fejezz be mindent, és csináld, amit én mondok, vagyis aktivizáld ezt és ezt a megszakítást —, tehát újra meghívódik az 1Ch, pedig még nem is fejeztük be az előzőt. Ez így mehet a végtelenségig.

Látszólag megfúrható a rendszer úgy, hogy a vírus esetleg rögtön az 1Ch megszakítást írja át, és ezek után bármihez hozzányúlhat, mivel már nincs az a rutin, amelyik ezt figyelné. Mi ezt úgy oldottuk meg, hogy az 1Ch átírását is figyeljük valahonnan; onnan, amit az 1Ch is figyel. Ha pedig már nem látják egymást, akkor tudjuk, hogy valaki garázdálkodik a gépnél. Nagyon sok program használja ezt a megszakítást, tehát nagy megszorítás lenne, ha ennek az átírását megtiltanánk. (A mágneslemezen találhat egy olyan programot az olvasó, mely az interruptvektorok elérésével és helyreállításával akadályozza meg a nemkívánatos beavatkozásokat a rendszerben.)

A sorozat folytatásában már a szűrő 2. lépcsőjének megoldásáról lesz szó.

Krokovay Károly—
Radványi Tibor

Sorozatunk a betűsorozatokról II.

Karakterek vektortranszformációval

A kiadványszerkesztő rendszerek többsége támogatja az ún. PostScript formátumot, amelyet elsősorban lézernyomtatók alkalmazásánál használnak. Erről szólva, e cikkben a DTP rendszerekben általában már rendelkezésre álló fontok manipulációiról lesz szó, míg a következő havi folytatásban az egyedi karakterek tervezéséről.

A PostScript kifejezést manapság egyre több helyen halljuk mint egy misztikus varázsszót, de értelmét még a számítástechnikában jártas szakemberek is csak körülírni tudják, pedig nem varázslás. A PostScript egy programozási nyelv, amely a megjelenítendő oldal összes grafikus elemét a nyomtató hardversajátosságaitól függetlenül írja le. A PostScript interpretert általában a nyomtatók ROM memóriájában helyezik el, ezzel is biztosítva a nyomtatók vezérlésének kompatibilitását. Bizonyos nyomtatókhoz azonban vásárolható olyan szoftver, amelyet a számítógép központi memóriájába kell betölteni, és ezen az interfészprogramon keresztül vezérelhetjük a nyomtatót PostScript-utasításokkal. Mindkét megoldásnak vannak előnyei és hátrányai a másikkal szemben. Ha az interpreter a nyomtatóban van, akkor a számítógép terhelése csökken, mert a bonyolult számításokat a nyomtató végzi. Ha olyan lézernyomtatót kívánunk venni, amelyik tartalmazza a PostScript nyelvet, akkor a nyomtató ára nagyon megugrik, mert nem csupán a ROM-ot kell megvásárolnunk, hanem az egész bővítőkártyát (amely tartalmazza a Motorola 68000-es processzort, és a legalább 2 Mbájtnyi RAM memóriát — ez a számításokhoz szükséges). Ezáltal alaposan megemelkedik a készülék ára.

A lassúság ára

Ha az interpreter a nyomtató ROM-jában van, ez nem jelenti azt, hogy az egyébként lassú nyomtatás gyorsabbá válik. Ugyanis annak ellenére, hogy ezekbe a nyomtatókba a legjobb processzorokat szerelik be (általában Motorola 68000-est), a processzor órafrekvenciája legtöbb esetben csupán 8 MHz-cel működik. A hasonló vagy ennél is nagyobb teljesítményű processzorral (például Motorola 68020-as vagy 68030-as) működő számítógépek 16 MHz-es, 32 MHz-es vagy ennél is magasabb frekvenciájú órajellel dolgoznak, ezért az interpreter működése jelentékenyen felgyorsul: gyorsabb lesz a nyomtatás. A számítógépbe betölthető szoftvereknek nagy előnye az is, hogy velük nem csupán a drága lézernyomtatókon tudjuk megjeleníteni a PostScript állományunkat, hanem az olcsó tűs nyomtatókon is. Hátrány viszont, hogy a viszonylagosan lassú nyomtatás alatt nem tudunk a számítógépen dolgozni, mert azt a számítási munkákkal leterheltük.

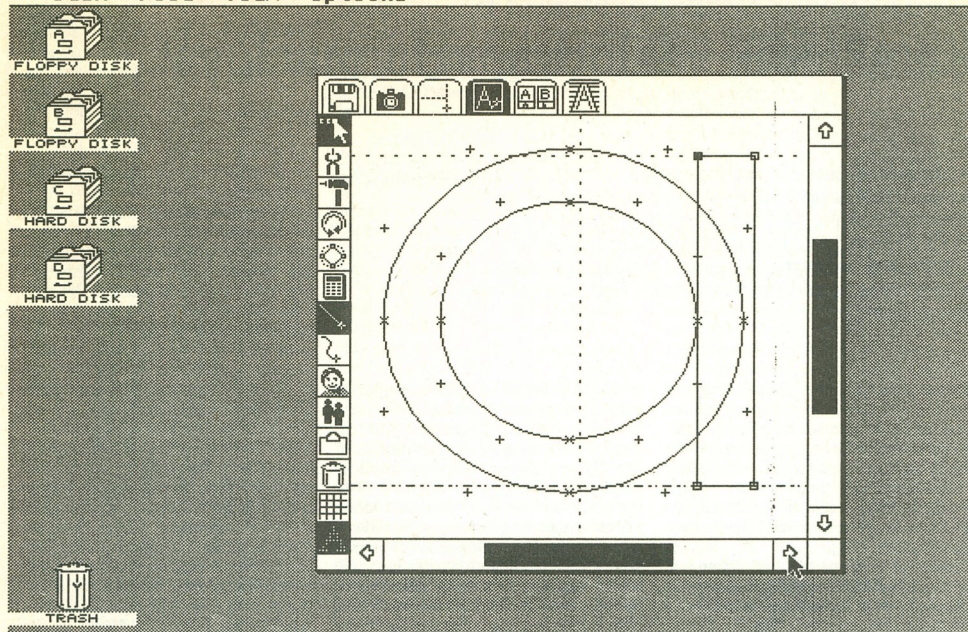
Annak ellenére, hogy a PostScript az utóbbi időkben — a DTP rendszerek elterjedésével egy időben — kezdett teret hódítani, mégsem mondható újnak. A hetvenes évek eredményeire támaszkodva John Warnock és Chuck Geschke fejlesztette ki a nyelvet az Adobe System Inc.-nél. Végső formájában a szabványosított nyelv 1982-ben készült el. A fejlődés ezzel nem zárult le, az Adobe cégnél már kifejlesztették a PostScript Level 2-t.

A PostScript utasításkészletével szinte az összes grafikai ábra leképezhető. A különálló ponttól a bonyolult görbékig minden elhelyezhető segítségével egy koordináta-rendszerben. A koordináta-rendszerben x és y irányban 1/72 inches lépéseket tehetünk, amely majdnem megfelel a tipográfiai alapegységnek, ugyanis 1 pont = 1/72,27 inch.

A PostScript más interpreterekhez hasonlóan ASCII-forrásszöveget dolgoz fel. A forráskód tartalmazza az oldal teljes leírását. Az interpreter soronként beolvassa a forrásszöveget, értelmezi az utasításokat, és feldolgozza a nyomtató számára. A PostScript interpreter kilenc betűcsaládot ismer különböző attribútumokkal, amely összesen 35 betűkészletet jelent. Az általánosan használt nyomtatókkal ellentétben a PostScript nem a betűkhöz tartozó bittérképet tárolja, hanem a betűt grafikus alapelemeire bontja, és a vonalakat, íveket és a szögeket tárolja le, rajzutasításként szerepeltetve a memóriában. Ennek nagy előnye, hogy a karakterek nagyításával, kicsinyítésével, megdöntésével nem torzulnak a betűk, a minőség ugyanaz marad. A bittérképével tárolt karakterek nagyításánál torzulást tapasztalhatunk, mert a karakter nagyításával nemcsak a mérete változik, hanem a bittérképben tárolt pontok is nagyobbak lesznek, tehát a kép raszteresebb lesz, a felbontás csökken. Ezt kerülhetjük el a PostScripttel. Ezenkívül másik előny, hogy a PostScripttel

Szótár

DTP: Desktop Publishing: számítógéppel támogatott kiadványszerkesztés.
Font: egy meghatározott méretű és stílusú, egy betűképből származtatott karakterkészlet. Például: 12 pontos Garamond book.
Méretezhető fontok: matematikai képletekkel leírt körvonal formájában tárolt fontok, amelyeket nyomtatás közben konvertálnak bittérképpé.
PostScript: az Adobe oldalleíró nyelve a megfelelő fontokkal. A PostScript printerekbe 35 fontot építenek; bonyolult grafikai effektusokra ad lehetőséget.



tetszőleges szövegkép alakítható ki, a szöveg és a grafika szabadon keverhető.

Vonzalmak és simulások

Az interpreter Bézier-görbékkel közelíti a megjelenítendő görbéket, egyenesekkel az egyeneseket. A Bézier-görbék a következő fontos tulajdonságokkal jellemezhetőek:

1. A Bézier-görbékhez tartozó alappontok — a kezdő- és a végpont kivételével — nem a görbén helyezkednek el. Ezek az alappontok mégis egy bizonyos „vonzó" hatást gyakorolnak a görbe hozzájuk eső részére. Ha az első és az utolsó támpont egybeesik, akkor a görbe zárt.

2. A Bézier-görbe független az alappontok helyzetének meghatározására szolgáló koordináta-rendszertől.

3. Ezek a görbék csak globálisan változtathatók. Bármely alappont elmozdítása a görbe minden részét megváltoztatja.

4. A Bézier-görbék hullámzáskiegyenlítő tulajdonságúak. Belül fekszenek az alappontok konvex burkoló sokszögén. A görbék ingadozása az alappontjaik körül nem jelentős.

5. A felhasznált alappontok számától függ a görbék variálhatósága és változtathatósága.

6. A bonyolultabb alakzatok elkészítése több egyszerű Bézier-görbe összekapcsolásával oldható meg. Ekkor azonban biztosítani kell a görbék közötti átmenetek folytonosságát. Ezt több, egymásra helyezett alappont segítségével valósíthatjuk meg. A Bézier-görbék előállításához a következő képleteket használhatjuk:

$$x(u) = \sum_{i=0}^{n} x_i B_i, n(u) ,$$

$$y(u) = \sum_{i=0}^{n} y_i B_i, n(u) ,$$

ahol $B_i, n(u)$ az ún. súlyfüggvény, definíciója:

$$B_i, n(u) = \binom{n}{k} u^i (1-u)^{n-1} .$$

Ezek a képletek egy $n+1$ alappontú Bézier-síkgörbe paraméteres egyenletei, ahol x_i, y_i az alappontok koordinátái, az u paraméter 0 és 1 között változik. Erre a paraméterre nézve a függvények n-ed fokú polinomok. Az ábrán egy „a" betű látható, melynek tervezésénél a fenti képletet használták. A és B a görbe alappontjait jelzi.

A PostScript ezzel a képlettel dolgozik. A standard PostScript interpreter 1000 pont méretű Bézier-körvonalainak az adatait tartalmazza, ez elegendően részletes bármilyen méret előállításához. Egy meghatározott méret előállításához a program a Bézier-körvonalat 0.001-gyel megszorozza, így megkapjuk az 1 pont nagyságú fontot, ezt pedig a PostScript értelmező bonyolult módon állítja be a kívánt méretre. Ez nem egy egyszerű nagyítás, annál jóval összetettebb módon működik az algoritmus, hogy jó minőségű betűket kapjunk.

A memóriában nem csupán a Bézier-körvonalakat tárolják, hanem egyéb, a karakterre vonatkozó információkat is, mint például a karakter szélességét. Ezzel lehetővé válik a proporcionális betűk nyomtatása. Ez azt jelenti, hogy a betűk különböző szélességűek: az i keskenyebb, mint például az m.

Kovács P. Attila

ALAPLAP

ÖSSZESÍTETT TEMATIKUS TARTALOMJEGYZÉK

1990. JÚNIUS — 1991. DECEMBER

A) SZERKESZTŐSÉGI ANYAGOK A NYOMTATOTT LAPBAN

ADATBÁZIS-KEZELŐK

ADATHORDOZÓK

CAD/CAM

CÉGINFORMÁCIÓK

DTP

EMBER–GÉP KAPCSOLAT

FELHASZNÁLÓI PROGRAMOK

GRAFIKAI PROGRAMOK

HÁLÓZATOK

HARDVER

HÍREK

IRODATECHNIKA

JÁTÉKISMERTETŐK

KÁRTYÁK

KIÁLLÍTÁSOK ÉS KONFERENCIÁK

KÓDKIOSZTÁS, KONVERZIÓ

KOMMUNIKÁCIÓ

KÖNYVISMERTETÉS

MÁSOLÁSVÉDELEM

MIKROBAZÁR

MULTIMÉDIA

MŰVÉSZETI ALKALMAZÁSOK

NYELVÉSZET

OKTATÁS

OPERÁCIÓS RENDSZEREK

PIACI INFORMÁCIÓK

PROCESSZOROK

PROGRAMOZÁSI NYELVEK

PROGRAMOZÁSTECHNIKA

REJTVÉNY

SEGÉDPROGRAMOK

SHAREWARE PROGRAMOK

SOLARSOFT SIKERLISTA

SZABVÁNYOSÍTÁS

SZÁMÍTÁSTECHNIKAI KULTÚRA

SZERKESZTŐSÉGI HÍREK

SZERZŐI JOG

SZOFTVERKATEGÓRIÁK

SZOFTVERÚJDONSÁGOK

SZOFTVERISMERTETÉS

SZÖVEGSZERKESZTŐK

TÁBLÁZATKEZELŐK

TANÁCSOK

TERMÉKPALETTA

TÖMÖRÍTŐK

UTILITYK

ÜZEMZAVAROK

VÍRUSOK

VONALKÓDTECHNIKA

VEGYES TÉMÁK

B) SZERKESZTŐSÉGI ANYAGOK A MÁGNESLEMEZ MELLÉKLETEN

C) HIRDETÉSEK

C/1) HIRDETÉSEK A NYOMTATOTT LAPBAN

C/2) HIRDETÉSEK A MÁGNESLEMEZEN

C/3) HIRDETÉSEK A LEMEZTASAKON

C/4) PLAKÁT, PROSPEKTUS

Összefoglaló tartalomjegyzékünket — a gyors visszakeresés érdekében — januári mágneslemez mellékletünkön is közreadjuk.

Menüzzünk!

A menükészítést támogató programokat négy fő típusba sorolhatjuk:

— A típus: COMMAND.COM-bővítések, melyek a menüket interaktív batchfájlokon keresztül hozzák létre.

— B típus: Primitív menüző, korlátozott számú programot tud elindítani, kevés paraméterrel. Ha paramétereket kell a menübe állított programoknak átadni, ezt főleg batchfájlokon keresztül tehetjük.

— C típus: Igazi menüző program. Egyszerű használat, rugalmasan és mélyebb DOS-ismeretek nélkül bővíthető. A létrehozott menük adatállományait saját (egyedi) formátumú adatfájljai(k)ban tárolják.

— D típus: Elsősorban DOS keretprogramok, használható menüzési lehetőségekkel.

Vannak kereskedelmi forgalomban lévő programok is, melyek a fenti négy típus valamelyikébe sorolhatóan támogatják a menükészítést, de ezek ára meghaladja az átlagos magánfelhasználók anyagi lehetőségeit. Van két program, amely annak ellenére, hogy nem szabad szoftver, szinte korlátok nélkül elterjedt már: a PathMinder (v2.08) és a Norton Commander (v3.0). Mindkettő rendelkezik menükészítő lehetőségekkel, melyek ismertetése az alábbi kiadványokban megtalálható:

1. Az IBM PC-ről kezdő felhasználóknak II., Szoftver (PathMinder, NC), LSI-ATSZ

2. Bartha Attila: A NORTON (Norton Commander), LSI-ATSZ

Más, kifejezetten menükészítést támogató programokról egy másik írásban fogunk részletesebben beszámolni.

A SolarSoft programkönyvtár természetesen a menükészítésre fogható programok széles választékával áll az érdeklődő felhasználók szolgálatára. A kínálatból a menükészítésre leginkább ajánlható programlemezek az alábbiak (Lemezszám, lemeznév, zárójelben a menüzést támogató program neve, besorolás, leírás):

SolarSoft #015: EXTENDED BATCH LANGUAGE (A típus)

Segítségével interaktív batch fájlokat állíthatunk elő. A csomag tartalmaz egy MENU.BAT nevű menüminta-állományt is, amely nagy segítségünkre lehet a saját menük összeállításában. Bár teljesen kezdők is használhatják, nem árt, ha az új menüt építő felhasználó már egy kicsit megismerkedett a PC-vel és a DOS-szal.

SolarSoft #017: DOS MENU (B típus)

1987-ben készült ez a BASIC-ben írt program, melynek EXE formában elkészített változata van a lemezen. A DOSMENUE.EXE program német nyelvű, és ugyancsak német nyelvű a leírása is, mely kissé hiányos. A program összesen harminc menüpontot tud kezelni, ami már elfogadható mennyiség. Az egyes menüpontokhoz kilencsoros parancssorozatot lehet rendelni, de ebben természetesen BAT fájlok meghívása is megengedett. A programot a DOSM.BAT-tal kell indítani.

SolarSoft #018: AUTOMENU 4.7 (C típus)

Az 1991. májusi Alaplapban a program egy előző változatát ismertettük. Az új, 4.7-es verzióban az előzőhöz képest számottevően a menü szerkesztését segítő AUTOMAKE program bővült. Most már, bár továbbra is hasz-

MenuMaster

Az új magyar shareware program alkalmazói komplett menürendszert építhetnek fel öt-hat függvény felhasználásával. Az egyes függvényekhez csak egy-egy paraméter tartozik. Kezelése akár negyedóra alatt is elsajátítható.

Redőnymenüket és vonalmenüket teljesen hasonló módon alakíthatunk ki. A menük tervezését követően egyetlen gombnyomással létrehozhatjuk a megfelelő Pascal, illetve C nyelvű forrásprogramokat, ezek a megszokott módon fordíthatók le, szerkeszthetők össze. A MenuMaster által létrehozott név.pas kiterjesztésű programok problémamentesen lefordíthatók a Turbo Pascal 4.0, 5.5 és 6.0 verzióival. Ahhoz, hogy egy tetszőleges Pascal programból használni tudjuk ezeket a menüket, be kell írni a főprogramba a "uses név"; direktívát.

A tervezés kezdetén a MenuMaster Create opciójából Create popup menus-t választva egy új redőnymenübe lépünk, ez már a tervezési menü. Ajánlatos ennek összes pontján végigmennünk kilépés előtt, csakis ebben az esetben kapjuk meg garantáltan a kívánt eredményt.

Névadással kezdődik a tervezés, alapértelemezésben rollxx a menünév, ahol xx egy egész szám. A program ellenőrzi, hogy nem szerepelt-e már előzőleg ez a név, illetve nem egyezik-e meg valamelyik Pascal vagy C kulcsszóval.

Helyes név megadása után a menü címe (fejszöveg), majd pedig a méret beállítása következik, végül elhelyezzük a képernyőn a menüt. Színezéskor megadhatjuk a háttérszínt, a szöveg színét, a sávkurzor háttérszínét, és az ebben megjelenő betűk színét, végül a hotkey-ként definiált betűk színét. Az egyes szövegek beírása egy egyszerű szövegszerkesztő segítségével történik.

A keretek ötféle karakterből képezhetők: üres karakter, egyszerű vonal, kettős vonal, csillagok, teljes karakter méretű vastag keret.

A View opció használatával ellenőrizhetjük a megtervezett menüt. Az így kialakított menükről kérhetünk további információt. Törölhetjük, ha rosszul sikerült, visszaléphetünk a létrehozó menübe.

Négy típusú árnyék közül választhatunk : nincs árnyék, fekete háttér — az előtérszín változatlan, fekete háttér — világosszürke előtérszín, fekete háttér — fekete előtér.

A menük aktuális állapota kimenthető egy név.mpr nevű állományba. A megtervezett és kimentett menürendszer a továbbiakban bármikor betölthető. **Cserna Csabáné**

nálhatjuk kedvenc szövegszerkesztőnket a menük (új és/vagy meglevő) szerkesztésére, sokkal inkább ajánlhatom az AUTOMAKE programot, amely minden segítséget megad a munkához.

SolarSoft #078: THE NAVIGATER (C típus)

A NAVIGATER lehetővé teszi EXE, COM vagy BAT állományok indítását a menüből. 34 program építhető így be, és mindegyiknek lehet 5 alkönyvtára. A NAVIGATER naplózza saját működését, és lehetővé teszi a menübe felvett programok jelszóhoz kötött használatát is. Extra szolgáltatása, hogy lekapcsolja a képernyőt, ha öt percig nem nyúlunk a géphez. A programhoz tartozik az installáló segédprogram is, de belülről is lehet konfigurálni. A programba online helpet is beépítettek. Hardverigénye 256 K és egy floppy (merevlemez, egér és printer használható, de nem elengedhetetlen).

Telepítése: NVGINST.

SolarSoft #094: PUSHBUTTON (C típus)

Kifejezetten menüző program, használható installáló segédprogrammal. A főmenüben 26 menüpont lehet, amelyekből szintén 26-26 pontos almenüket hívhatunk meg, így összesen 676 programot illeszthetünk be a PUSHBUTTON programmal kezelt menürendszerbe.

Jelszóval is titkosíthatjuk az egyes menüpontokat.

A program kezelése egyszerű, de nem árt hozzá ismerni egy kicsit a DOS-t és előkészíteni a menübe építendő programokat.

SolarSoft #341: ARGAFIND & ARGAMENU (B típus)

A lemezen egy gyengécske menüző program van egy jópofa másoló és egy használható állománykereső segédprogram mellett.

SolarSoft #354: PROBAT (A típus)

Profi módon megírt interaktív menüket tud előállítani.

SolarSoft #362: ANALYZE & X-BATCH (A típus)

Profi módon megírt interaktív menüket tud előállítani.

SolarSoft #M022: BEX (A típus)

Egy igazi magyar nyelvű, magyar fejlesztésű batchbővítés, amely nemcsak interaktív menük készítésére alkalmas, hanem a BAT fájlokat átalakítja EXE és DAT állománnyá. Ennek az az előnye, hogy a már rendszerint fáradságos és aprólékos munkával előállított menübe nem tud a kíváncsi és általában hozzá nem értő felhasználó belepiszkálni. Néhány dologban jobb a Norton Utilities Batch Enhancer programjánál.

SolarSoft #144: Point & Shoot (B típus)

Hiányzik egy PASBR.EXE állomány. Nem működött.

SolarSoft #145: DOS Controller (B típus)

Nem működött, monogépeken hibajelzéssel megáll már indításkor.

SolarSoft #234: Easy Access (B típus)

Ígéretes menürendszerként indul, de körülményes a használata az AUTOMENU és a NAVIGATER után, nem beszélve a többi remek programról.

Nagy Gábor

Játsszunk ismét!

Még a novemberi Alaplap játékdömpingje után is maradt némi tartalékunk. Hogy a választék még ínycsiklandóbb legyen, kissé megkésett desszertként a Solarsoft programkönyvtár #463 számú lemezét tálaljuk fel, amely Windows-játékokat tartalmaz.

A legkülönfélébb játéktípusokból választhatunk, ezek mindegyike tömörített formában található a lemezen. Úgy tűnik, sokan annyira megkedvelték a Windowst, hogy napi munkájuk mellett még játszani is így szeretnek. A lemezen található játékok többsége ismerős már mindannyiunknak, itt a Windows-technika, az egérrel történő választás, vezérlés nyújt kényelmesebb körülményeket.

A BACKGAMN-t ketten játszhatják. Kiemelkedően jól sikerült a játék grafikája.

A TETRIS mindenkinek ismerős lesz.

A MAHJON színeivel és grafikájával kellemes látványt nyújt.

A NIM játékban a számítógép az ellenfelünk. 36 gyufaszálból kell elvennünk 1 és 6 közötti darabot. Az nyer, akinél az utolsó gyufa marad. Ez általában a gép. Csak addig érdekes ez a játék, amíg rá nem jövünk a stratégiára — nem igazán nehéz. Attól kezdve az nyer, aki kezd.

A TICTACTOE során is a gép az ellenfelünk. 3x3-as amőbajáték, elég primitív.

A SOLITAIR kártyajáték, 7 sorban találjuk a lapokat, a négy ászra kell kitennünk növekvő sorrendben.

A FIFTEEN a közismert szám-ti-li-toli, csak most nem kell kézben tartanunk, a képernyőn jelenik meg a számtábla, egérrel kell a helyére tennünk a számokat.

A BREAKOUT során téglákat ütünk ki a falból, egyre feljebb jutunk, míg az összes tégla elfogy. A sebességet beállíthatjuk magunknak.

A lemezen található játékok közül ez az egyetlen, amelyhez EGA vagy Hercules monitor szükséges.

A PUZZLE 4 összekavart képet jelenít meg, ezeket kell összeraknunk.

A DICE kockajáték.

A STARBASE készítője kifejezetten azért fejlesztette ki 10 éves fia számára a Missile Command alapján ezt a játékot, hogy a gyerek ezen keresztül ismerkedjen meg a Windows-jellegű alkalmazások alapelemeivel. Olyanokkal például, mint a redőnymenük, párbeszédablakok. Fejleszti a játék a gyerekek szem- és kézkoordinációját. Különböző szinteken játszható, az egészen kezdőtől a haladóig. Automata módot választva folyamatosan játszhatunk, csak nem vesszük észre, hogy véget ért az egyik játékmenet, azonnal kezdődik az új.

Jó szórakozást decemberben is!

Verebély Pálné

A hatvány alapja

Kalkulátor

Az új magyar shareware program, a CALC az igényes matematikus felhasználókon kívül az „amatőrök" érdeklődésére is számíthat. A számítógép iránt bizalmatlan felhasználó persze ezt is „ellenőrizheti" a saját zsebkalkulátorával.

A program az Alt-J billentyűkombinációval indul. Ekkor a képernyőn egy olyan ablak jelenik meg, amely a megszokott zsebszámítógépek billentyűzetét szimulálja. Ha nem vagyunk megelégedve az eredeti színkombinációval, tetszés szerint átszínezhetjük, ha már tetszik a látvány, elmenthetjük.

A zsebszámológép aktuális módját az ablak jobb felső sarkában, a MODE szó után írt információk jelölik.

a) Az első szó a műveletek típusát adja meg:
SET — beállítások
ARIT — aritmetikus
TRIG — trigonometrikus
STAT — statisztikus
LOG — logaritmikus
HYP — hiperbolikus

b) a második szó az aktuális számrendszert adja meg:
DEC — tízes
BIN — kettes
HEX — tizenhatos

c) a harmadik szó a szögek mértékegységét adja meg:
DEG — fok, perc, másodperc
RAD — radián
GRA — tizedes osztású fokok

Az alapértelmezés az ARIT + DEC + DEG, de ez tetszés szerint változtatható.

Az egyes billentyűket két-három módon lehet kiválasztani:

a) a kurzorbillentyűk és a <RETURN> segítségével;

b) közvetlen kiválasztás — a számítógép billentyűi egy 6x4-es táblázatban vannak elhelyezve. Minden billentyű kiválasztható egy két billentyűből álló kombináció segítségével (sor—oszlop);

c) közvetlenül is beüthetők azok a billentyűk, amelyeknek van megfelelője a számítógép billentyűzetén ('+', '-', '*', '/', számjegyek).

Ha a billentyű egy unáris operátornak felel meg, az eredmény azonnal megjelenik, más esetben csak a második operandus begépelése és az '=' megnyomása után.

Az 'INV' billentyű inverz módba kapcsolja a számítógépet. Ez természetesen csak a trigonometrikus és hiperbolikus függvényekre és az insert statisztikus függvényre értelmezett.

Találunk néhány olyan billentyűt is, amelynek értelmezése további magyarázatot igényel.

Tízes számrendszerben három megjelenítési mód lehetséges:
NORM — normális;
SCI — tudományos (a szám hatvány alakban van kiírva, a hatvány mindig háromnak a többszöröse). Például: 12345 = 12.345 E + 3;
EXP — exponenciális (a szám hatvány alakban jelenik meg, egész része egy és tíz között van).

A számok bevihetők exponenciális formában is: a már beírt szám után le kell ütnünk a P (power) billentyűt, és be kell írnunk a kitevőt, amely előtt előjel is állhat. A hatvány alapja mindig az aktuális számrendszer alapja. Ha például tízes számrendszerben dolgozunk, akkor az 1, p, –, 5-ös sorozat a 0.00001 számnak felel meg.

Az aritmetikus függvények paraméterei:

a) lg(y)x (az 'x' szám 'y' alapú logaritmusa): az első szám az x, a második az y;

b) x ('x'az 'y' hatványon): az első szám az x, a második az y;

c) x (az 'x' szám 'y' fokú gyöke): az első szám az x, a második az y.

Statisztikus függvények:

Az <ins> segítségével egy számot írhatunk be. Inverz módban ez a függvény kivesz egy számot a bevitt számok sorozatából. A függvénnyel a beírt számok átlagát kapjuk meg, (x) pedig a diszperziót adja.

Indításkor választhatunk, hogy a CALC programot a rezidens változatát akarjuk-e használni. A rezidens változatot DOS 5-ös verzió esetén ajánlatos valamilyen más programból, például a Norton Commanderből indítani.

Cserna Csabáné

LIPROB

A számítástechnika oktatása során valószínűleg mindenütt felmerül, hogy az egyes programozási nyelvek használatát többé-kevésbé elsajátítják ugyan a hallgatók, arról azonban nincs pontos képük, hogyan lesz az általuk megírt programokból a számítógép számára is értelmes utasítások sorozata. Ehhez olyan programnyelv kellene, amely elég egyszerű (ez a könnyű megtanulhatóságot jelenti), ugyanakkor viszonylag rövid és érthető kódot generál.

A fenti követelményeknek próbál megfeleni a LIPROB nyelv. Szintaxisának tanulmányozásakor megállapíthatjuk, hogy egy Pascal-C-Basic hibriddel van dolgunk. A utasítás például Pascal-szerű, a kifejezések kiértékelésének módja a C nyelvből ismerős, az input utasítás pedig Basic jellegű.

A LIPROB interpreter számára a program egy virtuális számítógépen fut, amelynek gépi kódja igen egyszerű, az ember számára is érthető. Ennek ismertetése megtalálható a dokumentációban, érdemes alaposan áttanulmányozni. Ne ijedjünk meg, mindössze 24 utasításról van szó!

A program többféle módon használható:

a) futtathatunk egy LIPROB programot;

b) a generált kódot kimenthetjük (olvasható formában) lemezre;

c) betölthetjük és futtathatjuk a már kimentett kódállományt.

Ez utóbbi lehetőséget úgy is használhatjuk, hogy mi szerkesztünk meg egy kódállományt, és azt futtatjuk. Tehát egy programban két interpretert is kaptunk! Mielőtt azonban ezzel megpróbálkoznánk, ajánlatos némi jártasságot szereznünk a LIPROB kezelésében.

A magyar shareware lemezen külön könyvtárban találhatók meg a példaprogramok, nagyon tiszta logikával felépítve, didaktikusan haladnak az egészen rövid, pici programoktól lépésenként az egyre bonyolultabbak felé. Minden programhoz megfelelő mennyiségű magyarázó szöveg tartozik. A mintapéldákból és a dokumentációból bárki számára világossá válik (szinte programozási nyelvtől függetlenül), hogyan kell a változókat jól definiálnunk, milyen jellegű utasításokkal dolgozhatunk, hogyan érdemes függvényeket meghívni, illetve saját függvényeink megírásakor mire kell különösen ügyelnünk. A LIPROB támogatja a rekurzív programozást is.

Gépi kódú programok szerkesztése sem okoz gondot. A rendszer sokkal kevesebb ellenőrzést végez, amikor egy már elkészített kódot futtat. Ez azért van, mert alapértelemezésben LIPROB nyelven írt programokat futtatunk, az ezeknek megfelelő kód pedig jó. Így fokozott óvatosság ajánlatos a „gépi kódú" programok írásakor. A legjobb módszer az, ha alaposan tanulmányozzuk az egyes LIPROB programokból generált kódot!

A PELDAK.COD könyvtárban egy, kódban megírt rövid program található.

A lemezen külön könyvtárban található a LIPROB angol nyelvű és magyar nyelvű változata.

Mindazoknak ajánlhatjuk tehát a LIPROB magyar shareware lemezt, akik úgy szeretnének tanítani, hogy az eredmény ne káosz legyen, ne elmisztifikált fogalmakat szajkózzanak a tanítványok. Igazán nagy segítséget nyújthat abban, hogy egy életre megjegyezzék a gyerekek, nem ördöngösség, de nem is játék a géppel való kapcsolat, hanem olyan jellegű logikai lépések egymásutánja, mint amilyenek az elsőfokú egyenletek megoldásához szükségesek.

Verebély Pálné

SolarSoft sikerlista

Az 1991. július—szeptemberi eladások alapján

No.	Programnév	Db	Programleírás
1. 475	NEWSPACE	1	Harddiszkkapacitás-duplázó
2. 319	SCAN 7.6V80 & OTHERS	1	Vírusvédelem (McAfee)
3. 466	SKYGLOBE STAR GAZER	1	Mozgó csillagtérkép, PC-planetárium
4. 470	MULTI-EDIT 5.0	1	A legjobbnak tartott editor
5. 494	TEGLP WINDOWS TOOL	1	Ikongrafikus felület, ikoneditor
6. 480	GRASP 1.10C	1	Látványos animációkészítő program
7. 435	OPTIKS 2.18 & ICONVERT	1	PCX, PIC, GIF, TIF, GEM konverter
8. 510	ARJ 2.20	1	PKZIP/LHA-nál is hatékonyabb!
9. 418	FAST/SOFA/FFD	1	Új, szupergyors gépközeli nyelv!
10. 423	QFONT 1.5b	1	Fonteditor magyar Venturához
11. 495	TEGLC WINDOWS TOOL	1	Ikongrafikus felület, ikoneditor
12. 342	SPEECH	1	Tárrezidens, szövegfájlt felolvas
13. M24	BLISS	1	Főkönyvi rendszer C-ben!
14. 419	MODEL-S	2	Adatbázis-kezelő CASE DBASE-hez!
15. 429	QUEST 3-D 3.1	1	3 dimenziós mérnöki ábrázolás
16. 463	GAMES FOR WINDOWS	1	10+1 játék Windows 3.0 alá
17. 329	PC-MAGAZINE BENCH.	1	A szaklap hardvertesztjei 5.0
18. 383	4DOS V3.02	1	COMMAND.COM-pótló DOS-héj: 50 új parancs
19. 422	FONTEDIT 5.7	1	HP lézernyomtatókhoz szoftfontok
20. 432	LZEXE 0.91 & LIST 7.5e	1	Gyors EXE kompresszor, LIST PLUS
21. 417	UNISCREEN	2	Képernyőtervező C/PAS/BAS/FOR/COB/ASM/PRO/PRG!
22. 444	C-MIX #2	1	Rezidens programok készítése — C nyelv
23. M23	KEYBDRV	1	Teljes magyar ékezetesítő
24. 327	LHARC & LHICE	1	Adattömörítő — japán szuperprogram
25. 424	SSQL	1	Adatbázis-kezelő — SQL interpreter
26. 441	DATABASE IN C	1	C nyelv — adatbázis-kezelés (Btree, dBASE)
27. 443	C-MIX #1	1	EMS, XMS és egérkezelés C-ből!
28. 190	FINGER PAINT 4.0	1	PaintBrush-klón (CGA/EGA/HERC)
29. 168	BATTLE GROUND	1	2 személyes légicsata
30. 357	PEN PAL	1	Körlevélkészítő, címnyilvántartó

Apróbb eLHAjlások, avagy az LHA és a DR-DOS fasírtban van

A tömörítő programok rangsorában jó ideje előkelő helyet vívott ki Haruyashi Yoshizaki (Yoshi) LHARC (utóbb LHA) programja. A 2.10-es verziót már ismertettük az Alaplap júliusi számában, és a lemezmellékleten közre is adtuk a programot (az olvasók visszajelzései alapján közmegelégedésre). Az Alaplap Könyvek sorozat Tömör Gyönyör című kötetében pedig már a 2.12-es verzió titkaival is megismerkedhettek az olvasók.

A SolarSoft #509-es lemezén 2.13 az LHA soros verziószáma. A program fejlődése, szolgáltatásainak bővülése, megbízhatóságának növekedése szemmel láthatóan töretlen.

Mégis egy furcsa jelenségre figyeltem fel egy DR-DOS 5.0 alatt futó 4 megás 386-os AT gépen: nem tudtam arra rávenni az LHA-t, hogy az általam a parancssorban megadott sorrendben pakoljon be állományokat egy archívba, így a Telopot sem tudtam megfelelően elhelyezni. Emellett az adott könyvtárból nemcsak a megadott állományokat, hanem az összeset bepakolta. A végén ráadásul meg is hívta az egyik EXE állományt, amit a programmal csupán be akartam pakoltatni (történetesen a HYPER 2.50 programcsomag négy állományát, és a hozzájuk megszerkesztett (!) Telop fájlt akartam az LHA segítségével egy Telopos SFX-be begyömöszölni, mire az összes, a könyvtárban található állományt bepakolta, és a végén paraméterek nélkül elindította a HYPER.EXE-t.

Hiába próbálkoztam az újbóli tömörítéssel: LHA a HYPER25# ! HYP*.*, a tapasztalt jelenség nem változott.

Ijedtemben leteszteltem az egész rendszert a SCAN-nel, majd a többi hozzáférhető vírusdetektorral, de semmiféle vírust nem találtam.

Ezek után, vélvén, hogy ez a kópia sérült, megpróbálkoztam a birtokomban lévő összes egyéb LHA és LHARC verzióval, de azokkal is ugyanezeket a jelenségeket tapasztaltam.

Nem volt ezután meglepő, hogy otthon MS-DOS 3.30 alatt ketyegő 286-os gépemen mindegyik LHA és LHARC verzió teljesen hibátlanul dolgozott. A fentiek alapján egyértelmű volt, hogy a hiba nem az LHA-ban, hanem a DR-DOS 5.0-ban, vagy a 386-os gépben van. Ez utóbbi lehetőséget viszont ki kellett ejtenem, mikor egyéb DR-DOS 5.0 alatt futó (286-os) gépeken is ugyanazokat a zavaró jelenségeket tapasztaltam.

Milyen következtetéseket lehet levonni mindebből?

— Amelyik programról azt állítják, hogy teljesen hibátlan, azt biztosan még nem tesztelték le alaposan, talán még ki sem próbálták.

— Egy program fejlesztését nem befejezik, hanem abbahagyják.

Mit tehetünk, ha DR-DOS 5.0 van azon a gépen, ahol az LHA-t használni szeretnénk?

1. Cseréljük le MS-DOS 5.0-ra? Egy kicsit drága, munka- és időigényes ezért az apró kis hibáért.

2. Ne használjuk az LHA-t ? Kár lenne lemondani róla, hiszen nem az LHA-ban van a hiba.

A harmadik a nyerő változat: használjunk egy átmeneti könyvtárat az LHA-hoz, amelyet elnevezhetünk például \LHATMP-nek.

— Legelőször szerkesszük meg a Telopot ebben az \LHATMP könyvtárban, ha SFX-et akarunk előállítani.

— Ha szükséges, másodiknak a !.BAT indító állományt tegyük bele.

— Másoljuk bele az állományokat.

— Hozzuk létre először az archívot például az alábbi módon: LHA m Archív# ! !.bat *.*, csináltassunk belőle SFX-et: LHA s /x1 Archív#.

Ezután már csak a feleslegessé vált forrásállományokat kell eltakarítanunk, és az átmenetileg létrehozott LZH állományt.

Ha az LHAUTO program alkalmazásakor ezzel teljesen analóg módon kell eljárnunk:

— Telopot készítünk az \LHATMP-ben az SFX számára, és szükség esetén a !.BAT fájlt is elkészítjük, s elhelyezzük ide.

— A szokott módon beállítjuk az LHA változó értékét: SET LHA=/x1.

— Meghívjuk a programot: LHAUTO SFXfile ! !.BAT *.*.

— Sikeres bepakolás esetén töröljük a forrásállományokat az átmeneti könyvtárból.

— Helyezzük át a céllemezre a frissiben létrehozott SFX-et.

Nagy Gábor

Cégünk megbízható,
korrekt partner
a számítástechnikában

- Számítógépek és perifériák adásvétele
- ALR számítógépek
- OKI nyomtatók
- Szoftverek nagy választéka, oktatás
- WordPerfect-, Microsoft-, Ashton-Tate-termékek
- Hálózattervezés, -építés
- Átalánydíjas és eseti szerviz



1083 Budapest, VIII., Práter u. 51.
Telefon: 114-2696, 186-7836
Fax: 186-7836

INFORMÁCIÓKÉRÉS: 10 ▲

UNITRADE

Szervezési, kereskedelmi és Számítástechnikai K.F.T.

1073 Budapest VII., Erzsébet krt. 48.
Telefon/Fax: 142-2115

ŐSZI AKCIÓ!

Számítógép-konfigurációk:

AT 286 – 12/16 MHz, 1 MB RAM, 1,2 MB FDD, 44 MB HDD, Multi I/O, MGP, 14" mono monitor, Baby ház, 101 gombos billentyűzet....58 800,– Ft

AT 286 – 16/20 MHz, 1 MB RAM, 1,2 MB FDD, 44 MB HDD, Multi I/O, MVGA, 14" VGA monitor, Baby ház, 101 gombos billentyűzet....94 300,– Ft

AT 386 gépek már 130 000,– Ft-tól.

Alkatrészek:

VGA-monitor (1024x768, 0,28" képpont)....29 500,– Ft
MVGA-vezérlőkártya (1024x768, 512 kB RAM)..8 800,– Ft

A feltüntetett árak az ÁFA-t nem tartalmazzák.

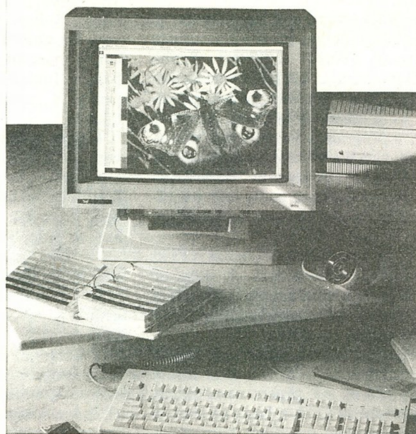
UNITRADE

...nem csak számítástechnika

INFORMÁCIÓKÉRÉS: 24 ▲

PC WORLD

magyarul



Az egész PC Világ a kezében, ha olvassa az **IDG Lapkiadó Kft.** 1992 januárjától megjelenő színes számítástechnikai magazinját,

a **PC WORLD**-öt,

amely az amerikai PC World és a német PC Welt magyar megfelelője.

Nélkülözhetetlen olvasmány

- az informatikai beruházásokért felelősöknek,
- a professzionális felhasználóknak,
- a megoldást kereső PC-tulajdonosoknak és amatőröknek.

A lap az új hardver- és szoftvertermékek és alkalmazások bemutatása mellett tanácsokat ad az eszközök kiválasztásához, hatékonyabb alkalmazásához.

Száz oldalon egy világot tarthat a kezében.

INFORMÁCIÓKÉRÉS: 27 ▼

Framework-world

Nem világvége!

Az előző számban
a korábban megszerzett ismereteinket finomítottuk:
a hibaérzékelő függvények alkalmazásával
az esetleges rossz adatokból bekövetkező abortálás
kerülhető el.
Egy Fred program futása a @trace segítségével
lépésről lépésre követhető; ez hasznos eszköz
a program kijavításához.
A saját, ún. felhasználó által definiált függvények
a táblázatok, adatbázisok kezelésénél
szinte nélkülözhetetlenek,
de a programjainkba is praktikusan beépíthetők.

Cikksorozatunk utolsó részében a Framework esetében tipikusnak tekinthető feladatok: táblázatok, adatbázisok feltöltése, lekérdezése, karbantartása esetén a folyamat megfogalmazását egyszerűsítő tartománykezelő (úgy is, mint tömbkezelő) függvények szerepelnek elsőként. A cikk második részében pedig az eddigieket végül is egésszé összefogó befejezés: hogyan készíthetünk menüvezérelt felhasználói rendszert? Ehhez a Fred-ben a megoldásmodulok (szubrutinok) valamiféle rendszerbe foglalásán kívül csupán néhány új függvény szükséges.

Tartománykezelő függvények

Az előzőekben programjaink készítésekor (az egyszerűség kedvéért) igyekeztünk elfelejteni, hogy a Frameworkben, azaz egy ügyviteli feladatok megoldására kifejlesztett integrált rendszerben vagyunk. A billentyűzeten beírt adatok legtöbbször táblázat- vagy adatbáziskeretbe jutnak, hogy rajtuk számításokat végezhessünk, meghatározzuk az adatok közötti összefüggéseket, s az aktuális állapotot grafikusan ábrázoljuk vagy nyomtassuk ki. Egy munkaügyi nyilvántartásból például különböző szempontok szerinti válogatást, statisztikákat kell készíteni, amihez már érdemes programot írni, hiszen rendszeresen ismétlődő feladatokról van szó. Ilyen esetekben adatok sorozatáról, adattömbökről beszélhetünk; ezek bevitele egy táblázatba vagy adatbázisba, majd a módosításuk: ez lesz most a témánk.

Pontosabban: a tartománykezelő függvények közé azok tartoznak, amelyek táblázat- és adatbázis-tartományok adatainak lekérdezésére, módosítására szolgálnak, illetve azok, amelyek lehetővé teszik ezt a tartomány első elemétől az utolsóig. De mi a tartomány? Ezt bizonyára minden olvasónk tudja: táblázatban vagy adatbáziskeretben kijelölt, téglalap alakú, összefüggő adatcsoport. Programban azonosítójával hivatkozhatunk rá. Eszerint kissé változik a kép: egyértelműen azonosítható a táblázat egy cellája, vagy az előbbi módon kijelölhető cellacsoportja; de: az adatbázis egy oszlopa (mezője), vagy több, egymás melletti oszlopa nem. A szükséges adatbázisrekordok kiválogatása a keretszegély képletterületén elhelyezett szűrőképlettel történik, azonosítóval ez nem megy.

A Framework interaktív használata során az adatbáziskeretben használt, ún. definiáló képlettel lehetséges minden rekordban ugyanazon mező értékének meghatározása. Ezzel az egyetlen képlettel tehát egy tartomány adatait adjuk meg.

Cikksorozatunk 2. részében a vezérlésátadó függvényeknél találkozhattak a @select függvénnyel, amely egy index alapján a paraméterlistából kiválasztott kifejezés végrehajtásával valósítja meg a többfelé ágazást. A @select a kifejezéseket egydimenziós tömbként kezeli, s az index adja meg, melyiket hajtsa végre. Ottani példánknál említettük: a stringsorozatból való kiválasztás történhetett volna a **@choose** függvénnyel is. A @choose első paramétere az index, az ezt követő listán azonban adatok vagy adatokat tartalmazó tartományok azonosítói lehetnek. Ezeket az adatokat a @choose egyetlen tartománynak tekinti, s az index alapján a megfelelő elem eszerint lesz kiválasztva. (Kétdimenziós tartományban a keresés sorrendje: balról jobbra, majd fölülről lefelé.) Például:

```
@choose(4,tabla.a1:tabla.b5)
```

eredménye tabla.b2 értéke. Hasonlóan

```
@choose(4,3,"alma",tabla.a1:tabla.b5)
```

tabla.b1 értékét adja eredményül. Teljesen mindegy, hogy a lista milyen típusú értékeket tartalmaz.

Ha egy táblázat- vagy adatbázis-tartomány adatait rendre le akarjuk kérdezni, a @choose függvénnyel a következőképpen tehetjük:

```
;ch
@local(index,adat,str,folyt),
index:=1,
folyt:=#true,
adat:=@choose(index,tabla.b3:tabla.d3),
@eraseprompt,
@while(folyt,
str:=@if(@iserr(@value(adat)),
@integer(adat),
adat),
@prompt("A tartomány
"&@integer(index)&". adata: "
&str,25),
@nextkey(2),
@eraseprompt,
index:=index+1,
@if(@iserr(@choose(index,tabla.b3:tabla.d3
)),
folyt:=#false,
adat:=@choose(index,tabla.b3:tabla.d3)))
```

Az index értékét (a kezdeti, cikluson kívüli beállítás után) a ciklusban növeljük mindaddig, amíg meg nem haladja a tartományelemek számát. Ha a tartományon túlra mutatna az index, a @choose függvény eredménye a **#value!** hibakonstans. @if függvénnyel figyeljük, hogy ez az állapot mikor következik be, s akkor a ciklus kilépési feltételéül szolgáló **folyt** logikai változó értékét #false-ra állítjuk. A lekérdezés adatbázisban is ugyanígy zajlik, persze a táblázattartományra való hivatkozás helyett adatbázis-tartományhivatkozást kell használni (például: BOLT1.költségek, a **költségek** mezőnév).

Ugyanezt a **@get** függvénnyel is megoldhatjuk. A @get a tartomány egy láthatatlan pointerrel kijelölt elemének aktuális értékét szolgáltatja. A pointer

a programban először szereplő @get-nél a tartomány első elemére mutat, a következő elemre a **@next** függvénnyel mozgatható. A pointer nem mozgatja a kurzort, tőle teljesen független. A @get függvény értéke akkor #NULL!, ha a tartományban már nincs további cella. Az alábbi módosításban az **index** változót csak a kiírandó üzenet miatt hagytuk meg, a keresésben nincs szerepe.

```
;getfv
@local(index,adat,str,folyt),
folyt:=#true,
index:=1,
adat:=@get(tabla.b3:tabla.d3),
@eraseprompt,
@while(folyt,
str:=@if(@iserr(@value(adat)),
@integer(adat),
adat),
@prompt("A tartomány
"&@integer(index)&". adata: " &str,25),
@nextkey(2),
@eraseprompt,
index:=index+1,
@next(tabla.b3:tabla.d3),
@if(@get(tabla.b3:tabla.d3)=#null!,
folyt:=#false,
adat:=@get(tabla.b3:tabla.d3)))
```

A **@fill** egy tartomány elemeit tölti fel a megadott, bármilyen típusú értékkel, vagy ha harmadik paramétert is használunk, a növekménnyel rendre növelt értékkel:

```
;fillfv @fill(db.sorszám,1,1)
```

Ezen függvény működéséhez természetesen kell egy **DB** nevű adatbázis, benne egy **sorszám** mezővel.

A következő program a táblázat feltöltéséhez az adatokat egy adatbázisból olvassa. Ehhez előzőleg betölti az ADATB adatbázis- és a TTABLA táblázatkeretet. A táblázatba való adatbetöltéshez a **@put** függvényt használjuk. Ennek a tartományban történő továbbléptetéséhez is egy @next függvény szükséges.

```
;olvas-tolt
@echo(#off),
@isabend(#false),
@pk("{ctrl-d}ga:\fred\adatb.fw3{return}"),
@if(@isabend,
@list(@echo(#on),
@eraseprompt,
@prompt("Az adatbázis nem tölthető
be!",20),
@return("adatb"))),
@pk("{ctrl-d}ga:\fred\ttabla.fw3{return}"),
@if(@isabend,
@list(@echo(#on),
@eraseprompt,
@prompt("A táblázat nem tölthető be!"),
@return("ttabla"))),
@echo(#on),
@while(@and(@get(adatb.név:adatb.összeg
)<>#NULL!,
@get(adatb.név:adatb.összeg)<>#N/A!),
@put(ttabla.B2:ttabla.D10,
@get(adatb.név:adatb.összeg)),
@next(ttabla.B2:ttabla.D10),
@next(adatb.név:adatb.összeg)),
@return("vége")
```

A táblázatba három összefüggő mező tartalma kerül a táblázat három oszlopába mindaddig, amíg van adat az adatbázisban. Ha üres mezőt talál, befejezi az áttöltést.

Ha az átviendő adatok az adatbázisban nem összefüggő tartományban helyezkednek el, más, ún. rekordorientált módszert kell alkalmazni. Ez utóbbi néhány különbséget mutat: a művelet az éppen kiválasztott rekord egy elemével (mezővel) kezdődik. A @next és a @reset segítségével a pointer **rekordról rekordra** is mozgatható. A cella- vagy mezőérték helyett @next és @reset #TRUE logikai értéket ad, ha a következő rekordra lépés sikeres, #FALSE-t, ha nem. Ilyenkor a @get és a @put függvényekben a hivatkozás megváltozik: tartomány helyett **adatbázisnév,mezőnév** formában adjuk meg a szükséges mezőt. Ha például az előző példában szereplő ADATB adatbázisból csak a **név** és az **összeg** mezőket akarjuk átvinni a táblázatba, az áttöltő programrészlet ez lehet:

```
;olvas2-tolt
@local(tovabb),
tovabb:=#TRUE,
@reset(adatb),
@while(tovabb,
@put(ttabla.B2:ttabla.B10,@get(adatb,név)),
@put(ttabla.D2:ttabla.D10,@get(adatb,össze
g)),
@next(ttabla.B2:ttabla.B10),
@next(ttabla.D2:ttabla.D10),
tovabb:=@next(adatb))
```

A programban a kiolvasás az első rekorddal kezdődik a @reset miatt, ami az első rekordra viszi a pointert. Az eljárás jellemzője az is, hogy minden mezőhöz szükséges egy @put-@get pár. Mezőnév helyett használható a mező sorszáma is: az első mezőé mindig az 1.

Ha egy programban vissza kell térnünk a tartomány kezdő eleméhez, akkor is a @reset(tartomány-hivatkozás) függvényt kell használnunk.

A tartománykezelő függvényekkel kapcsolatos programunk egy telefonregiszter. Ha a TELEFON és a DB kereteket betöltöttük, s elindítjuk a TELEFON keretre írt programot, megadva a nevet, akinek a telefonszámára van szükségünk, a képernyő alján megjelenik a telefonszám. Ha nincs a keresett név a nyilvántartásban, kérésre felveszi, s ehhez bekéri a telefonszámot is. Ezután természetesen lemezre menti az adatbázist (DB).

```
;telefon
@local(adatn,adatb,adatt,szurof),
@echo(#off),
@setselection("db"),
@pk("{ctrl-f}o{return}"),
adatn:=@inputline("Kinek a
telefonszámára van szüksége?"),
szurof:="@execute(db,név="""&adatn&
""")",
@szurof,
@if(@get(db.név)=adatn,
@list(@echo(#on),
@eraseprompt,
@prompt(adatn&  " telefonszáma:
"&@get(db.telefon),30),
@nextkey(3),
@return(adatn))),
@echo(#on),
@eraseprompt,
@prompt(adatn & " telefonszáma nincs a
nyilvántartásban! Felvegyem i/n?",10),
@echo(#off),
@nextkey,
@if(@or(@key={i},@key={l}),
@list(
@setselection("db"),
@pk("{in}{ctrl-home}"&
"{ctrl-c}r1{return}{out}{return}"),
adatt:=@inputline(adatn& "
telefonszáma?"),
@put(db.név,adatn),
@put(db.telefon,adatt),
@pk("{ctrl-return}"))),
@echo(#on)
```

Az ebben a példában is alkalmazott (l. szurof) **on the fly** technika lényege, hogy a kifejezést úgy írjuk meg, hogy a végrehajtandó függvényt stringként adjuk meg. Például a

```
@hide(@setselection)
```

kifejezés hibás (Reference expected), mert a @hide paramétereként egy keretnevet kell megadni, a @setselection értéke pedig string: az éppen kiválasztott keret neve. A fenti kifejezés helyesen:

```
@local(kif),
kif:="@hide(" & @setselection & ")",
@kif
```

Adatok keresése táblázatban, adatbázisban

Nem feltöltésre, hanem bizonyos adatok kikeresésére használhatjuk a praktikus **@vlookup**, **@hlookup** függvényeket. A keresés speciális, rendezett **táblázatban történik:** @vlookup esetében a paraméterként megadott tartomány első sora, @hlookup-nál az első oszlopa szigorúan monoton növő számsort kell tartalmazzon. A függvény ebben a számsorban keresi a paraméterlistán elsőként szereplő számmal legfeljebb megegyező értéket (azt, amelyik vele egyenlő, vagy a hozzá legközelebbi, nála nagyobb számot). A @vlookup értéke az ily módon kiválasztott sorban az az — egyébként tetszőleges típusú — érték, amelyet a harmadik paraméter mint oszlopeltolás határoz meg. @hlookup-nál ugyanez a kiválasztott oszlopban a megadott soreltolással meghatározott érték.

Alábbi példánk egy ADO elnevezésű, felhasználó által definiált függvény, amelynek egyetlen paramétere az adóalap, s ebből a megfelelő adótábla alapján kiszámítja a fizetendő adót.

```
;adofiz
@local(fizet),
fizet:=@hlookup(@item1,
adofiz.adotabla.a1:adofiz.adotabla.g3,2)+
(@item1-@hlookup(@item1,
adofiz.adotabla.A1:adofiz.adotabla.g3,0))*
@hlookup(@item1,
adofiz.adotabla.A1:adofiz.adotabla.g3,1)
```

A függvénydefinícióban szereplő **adotabla:**

0	55000	90000	120000	150000	300000	500000
0	.12	.18	.30	.32	.40	.50
0	0	4200	9600	18600	66600	146600

A következő program lottószámokat generál egy táblázattartományba:

```
;lotto
@local(n,i,adat,egyenlo),
egyenlo:=#true,
n:=1,
@put(lotto.B1:lotto.B5,@int(@rand*90+1)),
@next(lotto.B1:lotto.B5),
n:=n+1,
@while(@get(lotto.B1:lotto.B5)#NULL!,
adat:=@int(@rand*90+1),
i:=1,
@while(@and(i ,egyenlo),
@if(@choose(i,lotto.B1:lotto.B5)<> adat,
i:=i+1,
egyenlo:=#false) ;If vége
), ;belső while vége
egyenlo:=#true,
@if(i=n,
@list(@put(lotto.B1:lotto.B5,adat),
@next(lotto.B1:lotto.B5),
n:=n+1
)) ;If vége
) ;külső while vége
```

A számok ismétlődését a belső while-ciklussal kerüljük el. Ha egy már létezővel azonos számot generált a program, új számot kérünk. A tartomány pointerrel kijelölt aktuális eleme csakis akkor veszi fel a generált értéket, ha azzal azonos még nem szerepel a tartományban. A tartományelem feltöltését a @put függvény végzi, amely — a @get-hez hasonlóan — a pointerrel kijelölt tartományelembe teszi a második paraméterként megadott értéket, amely bármilyen típusú lehet. Mivel ugyanazt a tartományhivatkozást használja mindkét függvény, egyetlen @next kell a tartományban való továbblépéshez. A programot, mint a lemezmelléklet LOTTO keretének látható, a táblázatkeretre is írhatjuk.

A Framework III-ban jelent meg a **@DBLOOKUP** függvény, amely adatbázisban is lehetővé teszi a keresést. A paraméterek, de maga a folyamat is más: az első paraméterben a megkeresendő adatot adjuk meg, a másodikban azt, hogy hol, melyik mezőben kell megtalálni, végül a harmadik paraméter annak a mezőnek a neve, amely a meglelt adattal kiválasztott rekordban a szükséges adatot tartalmazza. Ez az adat lesz a függvény értéke. Az ADATB adatbázisnál maradva, ha **Horváth Péter** fizetését, amely az összeg mezőben található, 20%-kal emeljük, ezt a következő programrészlettel tehetjük:

```
;dbkeres
@local(van,dolgozo,plusz,regi,uj),
van:=#true,
@reset(ujfiz.a2:ujfiz.d10),
@while(van,
dolgozo:=@inputline("A dolgozó neve?"),
plusz:=@value(
@inputline("A fizetésemelés mértéke
%-ban?","0",,#true)),
regi:=@dblookup(dolgozo,"adatb.név","adatb
.összeg"),
uj:=regi*(1+plusz/100),
@put(ujfiz.a2:ujfiz.d10,
@dblookup(dolgozo,"adatb.név","adatb.név"
)),
@next(ujfiz.a2:ujfiz.d10),
@put(ujfiz.a2:ujfiz.d10,regi),
@next(ujfiz.a2:ujfiz.d10),
@put(ujfiz.a2:ujfiz.d10,plusz),
@next(ujfiz.a2:ujfiz.d10),
@put(ujfiz.a2:ujfiz.d10,uj),
@next(ujfiz.a2:ujfiz.d10),
@eraseprompt,
@prompt("Van még újabb név i/n?",25),
@nextkey,
@if(@or(@key={i},@key={I}),
van:=#true,
van:=#false))
```

Menükészítés

Az összetettebb programjainkat alkotó szubrutinoknak, az értékátadásokhoz szükséges globális változóknak a futáshoz kéznél kell lenniük: a megfelelő kereteket a biztonságos működés érdekében célszerű konténerkeretben tárolni. A Frameworkben egy outline nagyon egyszerűen felhasználható menürendszerként, beleértve a láncolt választási lehetőségeket és a magyarázó üzeneteket is. A menü megjeleníthető a képernyő alján egy sorban (l. HELP) vagy teljes képernyős formában. Ez a FRAMES menü Number Labels kapcsolójának állásától függ: amikor be van kapcsolva, akkor horizontális menüt kapunk a szerkesztősorban, ha viszont nincs, akkor vertikális, a teljes képernyőt elfoglaló menüt.

Többszintű menürendszer is kialakítható; nem célszerű azonban 2 vagy 3-szorosnál mélyebbet készíteni. Az outline szekciói képezik az első menüszintet, a szekciónevek jelennek meg menüpontokként. Ha egy szekciónak alszekciói is vannak, azok almenüként szerepelhetnek. A kiválasztott funkciókhoz tartozó programokat a legalacsonyabb szinten levő keretekre kell írni, amelyeknek már nincsenek további alszekcióik. Ezen keretek belsejébe írandó a menüpont kiválasztásakor az üzenetsorban megjelenő magyarázó szöveg. Ezt mindig még azelőtt kell beírni, mielőtt a keretszegély képletterületére írt programot futtatnánk.

A menüből való kilépéshez a @quitmenu függvényt használhatjuk, amely az ezt a menüt hívó programnak adja vissza a vezérlést. Rendszerint külön menüponthoz rendeljük ezt a funkciót.

Utolsó példánkban a menüvezérlésű program felépítését, működését mutatjuk be, s bár a programokat leírjuk, kövessék a fentieket a képernyőképeken is!

```
;sajatmenu
@sajatmenu.Választható funkciók
;sajatmenu.Választható funkciók
; ha teljes képernyős menü legyen:
@if(@not(@sense(#fn)),
; ha alsósoros menü legyen:
;@if(@sense(#fn),
@performkeys("{ctrl-f}n")),
;egy üzenet-képernyő megjelenítése:
@display(bemutat),
@nextkey(4),
;a menü hívása:
@menu(Választható funkciók),
;ide tér vissza a menüből való kilépés
után:
@eraseprompt,
@prompt("Vissza a menüből")
```

A **@display** függvénnyel hosszabb információk vagy tetszőleges más keret nyitható ki a képernyőre. Előzőleg természetesen már a memóriában kell lenniük, ami a konténerkeretben való tárolással automatikusan megoldódik. A menüképernyő nem szerkeszthető, viszont rendkívül egyszerű a bővítése vagy valamely menüpont törlése: a menüpontokat tartalmazó konténerkeretben létre kell hozni az újabb menüpontnak megfelelő keretet, illetőleg a feleslegeset törölni kell.

```
;sajatmenu.Választható funkciók.üzenet
@eraseprompt,
@PROMPT("Jó munkát kívánok!",35),
@nextkey(2)
;sajatmenu.Választható
funkciók.szövegkeret
@eraseprompt,
@prompt("Most egy üres keretet fogok
készíteni!",15),
@nextkey(2),
@performkeys("{ctrl-uplevel}{ctrl-c}e")
;sajatmenu.Választható
funkciók.számolótábla.szokásos
@eraseprompt,
@prompt("Alapértelmezés szerinti méretű
"&
"táblázatot tudok készíteni!",10),
@nextkey(3),
@echo(#off),
@performkeys("{ctrl-uplevel}"&
"{ctrl-c}w50{return}h100{return}s"),
@echo(#on)
;sajatmenu.Választható
funkciók.számolótábla.[5x10-es]
@eraseprompt,
@prompt("5 x 10-es táblázatot tudok
készíteni",10),
@nextkey(3),
@echo(#off),
```

```
@performkeys("{ctrl-uplevel}"&
"{ctrl-c}w5{return}h10{return}s"),
@echo(#on)
;sajatmenu.Választható
funkciók.számolótábla.tetszőleges
@eraseprompt,
@prompt("Akkora táblázatot tudok
készíteni, " &
"amekkorát csak akarsz! Várj csak!",10),
@nextkey(3),
@echo(#off),
@local(oszlop,sor),
oszlop:=@inputline("oszlopok
száma?","5",,#yes),
sor:=@inputline("sorok
száma?","10",,#yes),
@performkeys("{ctrl-uplevel}{ctrl-c}w"&
oszlop&"{return}h"&sor&"{return}s"),
@echo(#on)
;sajatmenu.Választható
funkciók.számolótábla.vége
@quitmenu
;sajatmenu.Választható
funkciók.adatbáziskeret készítése
@local(sor,oszlop),
@echo(#off),
oszlop:=@inputline("az adatbázismezők
száma?",
"5",,#yes),
sor:=@inputline("az adatbázisrekordok
száma?",
"10",,#yes),
@performkeys("{ctrl-uplevel}{ctrl-c}w" &
oszlop&"{return}h"&sor&"{return}d"),
@echo(#on)
;sajatmenu.Választható funkciók.kilépés
@quitmenu
```

A menüs felhasználói rendszerek készítésekor még nyilvánvalóbb a modulszerű építkezés előnye: az egyes funkciók önállóan megírhatók, tesztelhetők, s a végén már csak össze kell szerkeszteni, persze a paraméterek átadására gondosan ügyelve.

A mi világunknak sincs vége, de számos további, talán kevésbé lényeges lehetőség ismertetésére helyszűke miatt már nem kerülhet sor. Reméljük, hogy segítettünk megtenni az első lépéseket a Fred nyelvű programozásban, az első programok elkészítésében.

Az előző szám gyakorló feladatainak megoldása:

1. A korábbi KELTEZES nevű program felhasználásával:

```
;datum
@local(év,ho,h,hónap,nap),
@if(@itemcount=0,
@list(
év:=@mid(@date1(@date),@len(@date1(@
today))-3,4),
ho:=@mid(@date1(@date),@len(@date1(@
today))-8,3),
h:=@if(ho="Jan",1,
@if(ho="Feb",2,
@if(ho="Mar",3,
@if(ho="Apr",4,
@if(ho="May",5,
@if(ho="Jun",6,
@if(ho="Jul",7,
@if(ho="Aug",8,
@if(ho="Sep",9,
@if(ho="Oct",10,
@if(ho="Nov",11,
@if(ho="Dec",12))))))
)))))),
nap:=@mid(@date1(@date),1,2)),
@if(@itemcount<>3,
@list(
@eraseprompt,
@prompt("Csak paraméter nélkül, vagy 3
paraméterrel működik!",15),
@return("paraméterhiba")),
@list(
év:=@integer(@item1),
h:=@item2,
nap:=@integer(@item3)))),
hónap:=@select(h, "január", "február",
"március", "április", "május", "június",
"július", "augusztus", "szeptember",
"október", "november", "december"),
év & ". " & hónap & " " & nap & "."
```

2.

```
;nevhossz
@local(nev,hossz,n,folyt),
nev:=@inputline(
"Írd be a neved; ENTER leütése előtt
javítsd, ha kell!"),
;ha nincs adat
@while(nev=#n/a!,
@nevhossz, ;vissza az elejére
@return("vége")),
hossz:=@len(nev),
n:=1,
folyt:=#true,
;nincs-e szám a névben?
@while(folyt,
@if(@not(@iserr(@value(@mid(nev,n,1)))),
@list(
@eraseprompt,
@prompt("Csak betűkből álló nevet
fogadok el!",15),
;kilép, ha számot talált
@return("adathiba"))),
@if(n<hossz,n:=n+1,
folyt:=#false)),
;a név és a hossz kiíratása az üzenetsor
közepére
@eraseprompt,
@prompt("A "&nev& " név
"&@integer(hossz)&
" betűből áll",(80-(19+hossz))/2)
```

A Framework programozási nyelvét, a Fred-et ismertető „tanfolyamunk" végére értünk. Reméljük, e hasznos segédeszköz alkalmazásával a korábbi, a Framework integrált programmal „fapadosan" megkezdett interaktív táblázat-, illetve adatbázis-kezelési munkamódszert olvasóink elektronikus íróasztalán is felváltják a praktikus, kényelmes, programvezérelt megoldások. Ehhez kíván sok sikert:

Kóczy A. Judit

A minőség megéri az árát!

Amikor elkezdtük, nemcsak mi, hanem a számítógép-vásárlók többsége is kezdő volt. Valamennyien elkövettük azt a hibát, hogy elhittük: érdemes a legolcsóbb, még éppen működő gépekkel foglalkozni. Az évek során azonban rengeteg tapasztalatot szereztünk, és ma már tudjuk: a számítástechnikában sincsenek csodák. A minőségnek és a megbízhatóságnak ára van, s ha ezen spórolunk, az mindkettő rovására megy.



A fejlett piacgazdaságú Nyugaton azt tartják: a nagyon olcsó, bóvli termék a szegények adója. Aki mégis megveszi, rövidesen tapasztalja: rossz üzletet kötött.

A mindenáron való árcsökkentés ugyanis odavezet, hogy egyre gyengébb lesz a minőség, míg végül elérkezünk arra a pontra, amikor ez már nem éri meg a még oly olcsó árat sem. Mi nem akarjuk, hogy ügyfeleink később azt mondják: átvertük őket, egy vackot sóztunk rájuk, ezért inkább nem is forgalmazunk a legalsó árkategóriába tartozó számítógépeket és nyomtatókat. Úgy gondoljuk: egy átlagosan jó minőségre minden vevőnek joga van. Mi ezért nem tekintjük jelszónak, hogy nálunk **a minőség mindig megéri az árát.** Ez több annál, **ez a mi üzleti filozófiánk.**

De ez az ár nem feltétlenül magas — sőt! Először is az **IR** tudja, hogy sokféle igény, feladat és természetesen sokféle pénztárca létezik. Ezért többféle áron, többféle minőséget kínál, az átlagos felhasználónak megfelelő, olcsó „standard" típustól a profiknak szánt, előtesztelt alkatrészekből szerelt, 72 órán át járatott „extra" kategóriáig, amelyhez kétéves garanciát adunk.

De nemcsak ezért lehetséges, hogy az **IR Szervizben** mindenki megtalálja az igényeinek és anyagi lehetőségeinek megfelelő számítógépet. Segíti ezt az is, hogy az **IR** ma már akkora forgalmat bonyolít le, hogy szállítói megadják neki a legjobb vevőnek járó árkedvezményeket. Az **IR** pedig ezeket **a kedvezményeket maradéktalanul átadja** a hazai számítógép-felhasználóknak.

Így lehetséges, hogy egy kiváló minőségű 286-os AT, 40 Mbájtos merevlemezzel, 1 Mbájt RAM-mal, floppyval és nagyfelbontású, színes VGA monitorral **10%-kal olcsóbb az átlag hazai árnál.** Hasonló a helyzet a hordozható számítógépeknél is. De ugyanilyen jók az árai az egyedi igények alapján összeállított konfigurációknak és hálózatoknak is.

És ez még nem minden. **A nyári hónapokra időszakos számítógépvásárt hirdet az IR Szerviz.** Ennek során — konfigurációtól függően — akár **50 000 forintos árkedvezményt** is kapnak a vásárlók. Az oktatási intézmények pedig további nagyon jelentős, **25%-os kedvezményt** élveznek.

Természetesen a választás joga az Öné.
Keressen fel bennünket Budapesten,



IR Szerviz
VII. kerület, Kis Diófa utca 6.
Tel.: 121-3230, 141-0880



írjon levelet, vagy egyszerűen hívjon telefonon, a 121-3230 vagy 141-0880-as számokon, és mi segítünk, hogy kiválassza az Önnek megfelelő számítógépet. **Bármit választ, nyugodt lehet: olyan minőséget kap, amely biztosan megéri az árát!**

INFORMÁCIÓKÉRÉS: 06 ▶

Minden napra egy PC!

RAM és ROM a romokon

Ha valaki október elején bámészkodva sétált a Karolina úton, meg sem fordulhatott a fejében, hogy itt, a Budaflax hosszan elnyúló épülettömbjének belsejében új áruház nyitására készülnek. Közép-Európa legnagyobb számítástechnikai és irodatechnikai áruházának bejáratát, portál még két nappal az október 15-i nyitás előtt is alig lehetett észrevenni, gőzerővel dolgoztak az építők. A tervkészítéstől számított öt hónap(!) alatt mégis sikerült a fonóüzemből átépítéssel kialakítani az áruházat, s a Compfair '91-gyel egyidejűleg megnyitni azt — cáfolva minden előzetes borúlátást. A Cédrus Karolina Áruház november közepéig még nem sokat változott: kívül az elriasztó, romos homlokzat, az önmagukat álcázó építőkkel, belül az európai színvonalú belső tér és árukészlet.



Bod Péter Ákos ipari és kereskedelmi miniszter megnyitja a Cédrus Karolina Áruházat

Vásárló? Aki elég bátor volt, és gyanakodva bár, de bemerészkedett, nem bánta meg. A kínálatban — a Cédrus Rt. üzletpolitikájának megfelelően — a hangsúly a kiegészítőkön, kellékeken és a szoftveren van. Ezt egészíti ki új profilként a néhány gépcsaládra korlátozódó hardverkínálat, az irodatechnika és az irodaszer, valamint néhány telekommunikációs eszköz, számítástechnikai szakkönyv és külföldi folyóirat. struktúrába. Az új kínálatban mintegy 8000-féle termék szerepel, a legolcsóbb Staedtler-ceruzáktól a legdrágább fénymásolóig.

Hogy érdemes bejönni, és ott vásárolni, arról egy nem mindennapi bevezető reklámakció is igyekszik meggyőzni a potenciális vevőket: november vége óta ugyanis minden nap kisorsolnak egy Samsung AT-t azon vásárlók között, akik ezer forint feletti összegben vásárolnak. Az összeg nem mellékes, mert ezer forintonként jár egy sorsjegy, ami nagyobb beszerzés esetén igencsak megnöveli a nyerési esélyeket.

Az áruházat kiszolgáló részlegek folyamatos átadásának csúcspontjaként december közepére az áruház épületében megnyílik a PC Kocsma is, amely gasztronómiai funkcióján túl a számítástechnikai szakemberek találkozóhelye kíván lenni.

A Cédrus Karolina Áruház megnyitása nem jár együtt a jól bevezetett belvárosi Floppyland szaküzlet bezárásával, és azt sem jelenti, hogy a Cédrus önmagának csinál „vevőelszívó" konkurenciát. A szaküzlet és az áruház inkább erősíteni hivatott egymás forgalmát. A tömegáruk — szoftver, shareware, Polaroid és Perfect Data termékek — mindkét helyszínen megtalálhatók, de a Floppylandben a bolt mérete korlátot szab a választéknak.

Bár a Cédrus Karolina Áruház a kitűzött napi 2-3 millió forintos bevételi tervét — az építkezés viszontagságai miatt az első hónapban érthető módon — nem tudta elérni, de a dinamikus forgalomnövekedés prognosztizálható.

V. J.



Ennek az áruháznak hangulata van...



...ha kívülről ebből eleinte sok nem is látszott.

Hosszú utat tettél meg, PC!

Tíz évvel ezelőtt, 1981 augusztusában a nagyszámítógépes rendszerek éllovas gyártója piacra dobta első mikroszámítógépét. Az IBM tíz évvel ezelőtti értesítőjéből érdemes felidézni néhány részletet.

New York, augusztus 12. — *Az IBM cég ma bejelentette a legkisebb, legolcsóbb számítógépes rendszert, az IBM személyi számítógépet. Az üzleti, iskolai és otthoni használatra tervezett, egyszerűen kezelhető rendszer nagyon alacsony áron, 1565 dollárért kapható. Fejlett szolgáltatásai révén további programok megvásárlásával több száz népszerű alkalmazói programot tud működtetni.*

Az értesítő az új rendszer néhány kitűnő tulajdonságának leírásával folytatódik. Például: *akár 262144 karakteres felhasználói memóriája is lehet (alapkiépítésben 16384 karakter)*. Ez a 16384 karakter 16 kilobájtot jelent — ami abban az időben persze bőven elégnek tűnt. Az IBM jelenlegi asztali csúcsgépének, a PS/2 Model 90-nek RAM-ja 8 megabájtos...

Azután ott volt a „gyors, 16-bites mikroprocesszor ... amelynek műveleti sebessége a másodperc milliomodrészével mérhető." Ez a csip az Intel 8088-as volt, melynek regiszterei ugyan 16 bitesek, de adatsínje 8 bites — úgy tervezve, hogy minden PC néhány dollárral olcsóbb lehessen, és talán hogy a gyártók valamivel egyszerűbb bővítőkártyákat tervezhessenek. A 8088-as 4,77 MHz-en futott. Emlékszünk még, mekkora sebességnek tűnt ez akkor? Éppen mostanában jelentette be az Intel, hogy sikerült a 8088-as ük-ükunokájának sebességét (vagyis a 486-osét) egészen 100 MHz-ig felpörgetnie.

Az akkori új rendszer képes volt „256 karaktert a 16 előtérszín és 8 háttérszín bármelyikében megjeleníteni ... és 4 színű grafikával rendelkezett." Hogy ezeket a színeket megkaphassuk — a valóságban egyáltalán színeket kaphassunk —, vásárolni kellett külön egy grafikus kártyát, s ez olyan megoldásnak bizonyult, ami elindította a különböző bővítőkártyák későbbi lavináját.

Az újdonság megjelenését kísérő értesítő azt hirdette, hogy az IBM PC *tömör* gép. Hát... valóban kisebb volt, mint az IBM System 370-es nagygép. Manapság vehetünk már olyan laptop gépet, amelynek súlya az IBM PC billentyűzeténél is kisebb.

Az értesítő azt is írja, hogy az IBM PC-t egyaránt szánták otthoni és irodai használatra. Az egyik fénykép például két kisfiút ábrázol, amint a nappali szoba szőnyegén hasalva a MUSIC.BAS-sal játszanak a PC-n. Játszhattak volna akár a DONKEY.BAS-sal is, ami szintén a rendszerhez tartozott (s egyébként a Microsoft mai vezetője, Bill Gates készítette). Az új rendszeren használható alkalmazói szoftverek felsorolásában megtalálható a klasszikus Adventure kalandjáték Microsoft által készített változata.

Természetesen a Microsoft „fejlett lemezes operációs rendszert" szállított a számítógéphez. A biztonság kedvéért az IBM a PC-DOS mellett opcionális operációs rendszerként biztosította a CP/M-86-ot és az UCSD rendszert is. Az értesítő szerint: „ez a két rendszer lehetőséget teremt a felhasználóknak arra, hogy minimális módosításokkal a széles körben használt alkalmazások százait vigyék át az IBM személyi számítógépre".

Mondanunk sem kell, nem túl sok alkalmazás jelent meg ezekkel a kiegészítő operációs rendszerekkel, ezzel szemben megszámolhatatlanul sok új program született a külön megvásárolható Basic fordító felhasználásával, és a Microsoft elnöke azóta valóságos birodalmat hozott létre.

Ha visszatekintünk, megmosolyoghatjuk az eredeti IBM PC lehetőségeit. De azért gyengéd törődéssel tesszük ezt. Annak ellenére, hogy olyan rendszerek voltak a vetélytársai, mint a Macintosh és az erőteljes RISC gépek, a PC olyan számítógépek szülőanyja, amelyek kétségbevonhatatlanul vezető helyen maradtak a kisszámítógépek között.

Sajnos a tizedik évforduló megünneplésében nem vehet részt az az ember, aki az IBM PC fejlesztési csoportját vezette. Don Estridge, az IBM kezdőrendszerek (entry-systems) részlegének vezetője 1985-ben repülőszerencsétlenség áldozata lett. (A *kezdőrendszerek* elnevezés jól szemlélteti, hogy mi lehetett az IBM elképzelése a PC helyéről termékeinek sorában.)

(Byte, 1991/augusztus)

Mítosz, misztikum, mánia — Microsoft

Ha a Microsoft valóban olyan jelentős erő az iparban, akkor miért a többieket követi? Vajon elvesztette a lendületét? Úgy tűnik, a Microsoft még a felhasználóknál is homályosabban látja a jövőt.

John-Louis Gassee-nek a MacWeek-ben megjelent cikke azt sugallja, hogy az Apple multimédiát támogató agitációja valójában csak trükk, amivel félrevezetheti a „másolós" Microsoftot, és olyan területeken pazaroltathatja el vele erőforrásait, ami zsákutca. Kinek kell ugyanis a multimédia? Talán az ABC News hírműsornak. De ki az közülünk, akinek

gondosan összeállított, multimédiát alkalmazó előadásokra van szüksége?

A Microsoft nem egyszer követte már a vezető Apple-t. Az Apple grafikus felhasználói interfészt hozott létre, erre a Microsoft elkezdett a Windows-on munkálkodni. Az Apple a Mac-re Basic-et fejlesztett, és a Microsoft is ezt csinálta. Az Apple egere után hamarosan a Microsoft is egeret készített. Az Apple kifejlesztette az Apple Works-t, hamarosan láthattuk a Microsoft Works-t. Szinte biztos, hogy a Microsoft az Apple legutóbbi trükkjét is lemásolja: a mikrofonnal rendelkező, hangos jegyzetelő rendszert.

Ezek a térdreflex-szerű reakciók a piacon megjelenő újdonságokra a Microsoftnál megszokottá váltak. Angliában a Digital Research Inc. (DRI) bejelentette a DR DOS 5.0-t, és a Microsoft személyzete ott volt, hogy feljegyezze a rendszer jellegzetességeit. Nem sokkal utána, még mielőtt a DRI az US-ban is bejelethette volna termékét, a Microsoft a PC Week részére kiszivárogtatta, hogy DOS 5.0-t ad ki, majdnem azonos tulajdonságokkal. Nemrégiben pedig, éppen néhány héttel azelőtt, hogy a Go Corporation bejelentette toll-alapú operációs rendszerét, a Microsoft körbeszaladt az USA-ban, és bemutatta a toll-alapú Windows programot.

Néhány évvel ezelőtt a Sony és a Philips felvásárolta a CD-I szabványt, és hirtelen a Microsoft a DVI-vel csapott vissza, miközben megpróbálta megvásárolni a Microware Systems céget, a CD-I operációs rendszerének gyártóját. Egyik pániklépés a másik után. Az emberek elkezdenek a Unix-ról csevegni, erre a Microsoft kemény összegeket fektet a Unix szállítójába, a Santa Cruz Operation-be (SCO). Vajon melyik játékot találjuk a Microsoft által kiadott Windows játékcsomagban? A Tetris-t!

Bill Gates és vezető csapata magabiztos álláspontot képvisel, amikor a szakmát elemzi, viszont elgondolkodtató a cég határozatlan, egyik hóbortból a másikba tántorgó, majdnem paranoiás erőlködése. Ez a termékekre vonatkozó „én is!" zavarodottság kiterjed a csomagolásra, az elnevezésekre, de még a számozási konvenciókra is. A WordPerfect kiadja az 5.0-ás verziót, tehát a Microsoft Word is gyorsan 5.0-ás verziószámmal jelenik meg. A Lotus kiadja az 1-2-3 számolóprogram 3.0 változatát, s ezt nyomban követi a Microsoft Excel 3.0.

A Microsoft szalad minden után, amit meglát. Cselekedeteiben nincs előretekintés. Kergeti a toll-alapú rendszereket, a multimédiát, a Postscript-szerű lapleíró nyelveket, a CD-ROM-ot, a DVI-t, a könyvkiadást, egyszerre a grafikus és a karakteres alkalmazási felületet, a játékokat, a hálózatokat, stb. Vékonyra szétteríti erőforrásait.

Az ilyen típusú mánia az egyszeri sarki szatócsra emlékeztet: a bolt tulajdonosa egyszerre árusít szarvasagancsot és bizsut, vállalja a kenyérpirító javítását és fűnyírógépet élez, de van a boltban elektroncső-tesztelő és tekerős rágógumi automata is. A boltos felesége pedig állandóan attól fél, hogy férje a következő vásárról megint hazahoz néhány „biztosan és gyorsan meggazdagodsz" típusú szerkentyűt. Nem lenne jobb, ha inkább elküldené a férjét a Microsoft-hoz dolgozni?

(PC Magazine, 1991/április)

Lúdanyó CD-ROM-on

A Sierra On-Line „Mixed-Up Mother Goose" (Bolondos Lúdanyó) játékát három éve adták ki, és az olvasással éppen ismerkedő gyerekek körében azóta is nagyon népszerű. Az új verzió, melyet most egy csillogó kompaktlemezen szállítanak, nagymértékben különbözik valamennyi eddig ismert számítógépes játéktól. Ez az első beszélő kalandjáték, az első multimédia CD-ROM játék, amelyet tömeges használatra terveztek. A hangos-játékok megérkeztek!

Ahogy a 3 dimenziós, 256 színnel színezett Lúdanyó-világban kalandozunk, több különböző gyerekmondókát mondogató alakkal találkozhatunk, akik mind elvesztettek valamit. Látod azt a kislányt a kerítés mellett? Kattints csak az ajkakat ábrázoló ikonra, s a következő beszélgetést fogod hallani.

Picurka: „Báránykák! Hol vagytok, báránykák?"

Te: „Elvesztetted báránykáidat?"

Picurka: „Igen, és nem tudom, merre találom meg őket."

Te: „Segítek neked megkeresni báránykáidat."

Picurka: „Te biztosan az a jó kisgyerek vagy, akiről Lúdanyó mesélt már nekem."

Ezek után rád vár a feladat, hogy Picurka báránykáit megtaláld. Útközben sok más elveszett tárggyal találkozol, például az öreg Cole király pipájával, Humpty-Dumpty létrájával, Jack és Jill puttonyával, Miss Muffet székecskéjével.

Végülis a bégető báránykákat megtalálod. Kattints a báránykákra, erre azok hazakísérnek Picurhoz. Ő pedig olyannyira boldog, hogy azt mondja, „Az én báránykáim! És hogyan billegetik a farkukat! Nagyon köszönöm, hogy megtaláltad őket."

Ezek után elmondja kis versikéjét. Ez az első olyan szoftver, amelyet egy kórus kísér. Bár mindez túl szelídnek tűnik a gyilkoló sárkányok és a mindent széttipró űrszörnyek meséihez képest, a felnőttek ugyanúgy szeretik az ilyet, mint csemetéik.

A Lúdanyó teljes egészében ikonvezérelt. Miután a programot indító DOS parancssort beírtuk, a billentyűzetre már nincs szükségünk. Használhatunk botkormányt, hogy az ikonok között lavírozzunk, de mindez egérrel még kellemesebb. Olvasnunk sem kell — minden végrehajtandó utasítás szóban elhangzik — ez egy olyan játék, melyet az olvasni még nem tudók is egyedül játszhatnak.

A multimédia termékek szabványainak kidolgozása még a jövő feladata, ezért a Sierra két CD-ROM-ot is elhelyezett a csomagban. A piroscímkés lemez azoknak kell, akiknek CD-ROM lejátszójában már van hang. A korábbi CD-ROM készülékkel rendelkezőknek a kékcímkés lemez a megfelelő, amelyen a zajok és hangok digitalizáltak, és azok reprodukálásához további hangkártyát kell a gépbe helyeznünk. (Pl. SoundBlaster vagy Disney's Sound Source).

A piroscímkés CD hangminősége a digitalizált változaténál sokkal jobb, viszont a digitalizált változat egy további szolgáltatást is nyújt: mivel a digitalizált beszéd tárolásához sokkal kevesebb hely kell a lemezen, a Sierra cégnek sikerült a kék lemezre négy további nyelvi változatot is felvennie. Így a szereplők az angol mellett franciául, németül, spanyolul és japánul is megszólalhatnak. A nem angol szövegeket profi színészek mondták a lemezre, míg az angolokat maguk a Sierra dolgozói.

A viszonylag lassú CD-ROM egységeket ismerve azt várnánk, hogy a helyszínváltások lassúak, de ez nem így van. A Bolondos Lúdanyó ugyanolyan gyorsan és simán lejátszható, mint a merevlemezről futó játékok. Bár néhány kisebb fájlt merevlemezre installálhatunk, maga a program, a hanganyag és a grafika a kompaktlemezen marad.

Mint minden élenjáró technológiának, ennek is vannak hibái. A játék néha elveszíti a hangját, ha előzőleg másik játékot játszottunk. A Lúdanyó előtt ezért ajánlatos mindig rendszert indítani.

A Lúdanyó multimédia változatának az ára is megfelelő; pontosan annyiba kerül, mint a hajlékonylemezes változat (60 dollár). A Sierra ezzel is igyekszik elősegíteni, hogy a multimédia sokak számára elérhetővé váljon. Ugyanez a törekvés a CD-ROM egységek gyártóiról sajnos már nem mondható el. Ha a lejátszó készülékek olcsóbbak lennének, könnyen lehetne a Lúdanyóból a számítógépes korszak Hófehérkéje.

(PC Magazine, 1991/augusztus)

Javítják az ablakrepedéseket

A Windows 3.1-es előzetes kiadása számos új szolgáltatást és lehetőséget tartalmaz. Ezek közül némelyik hibajavítás, némelyik pedig meglepő újdonság. Bár még nem a végleges Windows 3.1 változatról van szó, lényeges eltérés már valószínűleg nem várható.

A Windows 3.0 egyik legtöbbet kritizált eleme a File Manager — tegyük hozzá, hogy jogosan. Ennek hiányosságai vezettek oda, hogy amikor fájlokkal kell dolgozni, sok felhasználó egyszerűen kilép a Windows-ból. Ez a 3.1-es verzióban megváltozik. Az új fájlmenedzser lehetőséget ad arra, hogy egyszerre több könyvtárat tekinthessünk meg, vagy akár egyszerre nézhessük meg a fastruktúrát, az ikonokat, a fájl jellemzőket vagy ezek valamelyik kombinációját. Ezen felül a korábbinál jóval több információt nyújt a lemezméretekről, a kötetcímkékről és hálózati kapcsolatokról is.

A Windows 3.1-hez hozzátartozik a TrueType. Ez egy intelligens betűkörvonal-készítő technológia, amely az Apple fejlesztése és a cég mostanában kiadott System 7.0-jában is megtalálható. A TrueType valódi WYSIWYG (amit látsz azt kapod) képernyő- és nyomtató fontokat szolgáltat a Windows részére. (A Microsoft dokumentációja szerint összesen 13 betűfajtát.)

A Microsoft Object Linking és Embedding (OLE) technológia fontos, előrevezető lépés a Windows 3.1-ben. Hasonlítható a Hewlett-Packard NewWave-hez és az Apple Publish/Subscribe-hoz a System 7.0-ban. Az OLE olyan mechanizmus, amellyel az adatok különféle megjelenési formáját (pl. grafikonok, táblázatok, adatbázis fájlok) egyetlen dokumentumban lehet összefogni.

Jelenleg, ha egy dokumentumba egy kívülről behozott elemet akarunk beilleszteni, általában a Clipboard-ot használjuk. Ha ez a behozott adat változik, mehetünk vissza és másolhatjuk ismét az új adatokat. Az OLE-t felhasználva a forrás-adat megváltozása nyomán a módosítás a Windows dokumentumban is automatikusan lezajlik. Ha pedig a dokumentumban rákattintunk egy ilyen behozott részre, azonnal a kapcsolódó alkalmazásban találhatjuk magunkat. A Windows 3.1 béta változatában a PC Paintbrush és a Cardfile támogatja az OLE lehetőségeket.

Az OLE lényeges lépést jelent majd előre, és az alkalmazások valódi integrálását teszi lehetővé, egy kicsit kiegyenlítve a Macintosh által az alkalmazások közötti kommunikáció terén jelenleg élvezett előnyt. Sajnos a meglévő Windows-alkalmazások többsége erről lemarad, mivel az OLE támogatást már a tervezésnél be kell illeszteni.

Hogy mit rejtettek a grafikus felhasználói felület mögé — ez valószínűleg a Windows 3.1 legmeglepőbb erénye lesz. Gyakori panasz a Windows 3.0-ra a sebesség hiánya, amikor nem 386-os, 486-os és rengeteg RAM-mal ellátott gépen futtatják. Ez a Windows 3.1-ben alaposan megváltozott. Meglepő sebességnövelést sikerült elérni!

A Windows 3.1-ben a fejlesztők szembetűnően sok munkát fektettek a visszafordíthatatlan alkalmazási hibák (UAE = unrecoverable application errors) megszüntetésébe. A Windows 3.0 felhasználók átka, az UAE, akkor jelentkezik, ha az alkalmazások nem várt paraméterekkel hívják meg a Windows-t, vagy, amikor különböző alkalmazások ugyanazt a memóriaterületet egyszerre próbálják használni.

Nagyobb sebességet ígér a FastDisk is, ami még újdonságnak számít a PC használói körében. Ez a virtuális eszközmeghajtó megkerüli a DOS-t és a BIOS-t, és közvetlenül a lemezegység kontrollerjével „beszélget". Védett módban működik a 386-os processzorokon, úgy, hogy az INT 13-as (hexa) merevlemezes BIOS hívásoktól elragadja a vezérlést. A FastDisk további előnye, hogy virtuális memórialapozó fájlt hozhatunk vele létre, ezáltal több nem-Windows alkalmazást is futtathatunk a háttérben, és közöttük kapcsolgathatunk.

A Windows 3.1 tehát a 3.0 változat miatti jogos kritikák legtöbbjét elcsendesíti. Nem lesz tökéletes, de nem is lehet az. A jelenlegi Windows verziók ugyanis nehéz terhet cipelnek: kifinomult burokba téve olyan öreg operációs rendszert csomagoltak, amelyet nem terveztek teljesértékű multitaszk szolgáltatásra. Ezt a hátrányt az IBM bizonyára kihasználja majd, mikor az OS/2 2.0-t piacra dobja. Másrészt a Windows 3.1 új tulajdonságai, a Microsoft-tól megszokott marketinggel fűszerezve, feltehetően további híveket toboroznak a Windows-használók táborába.

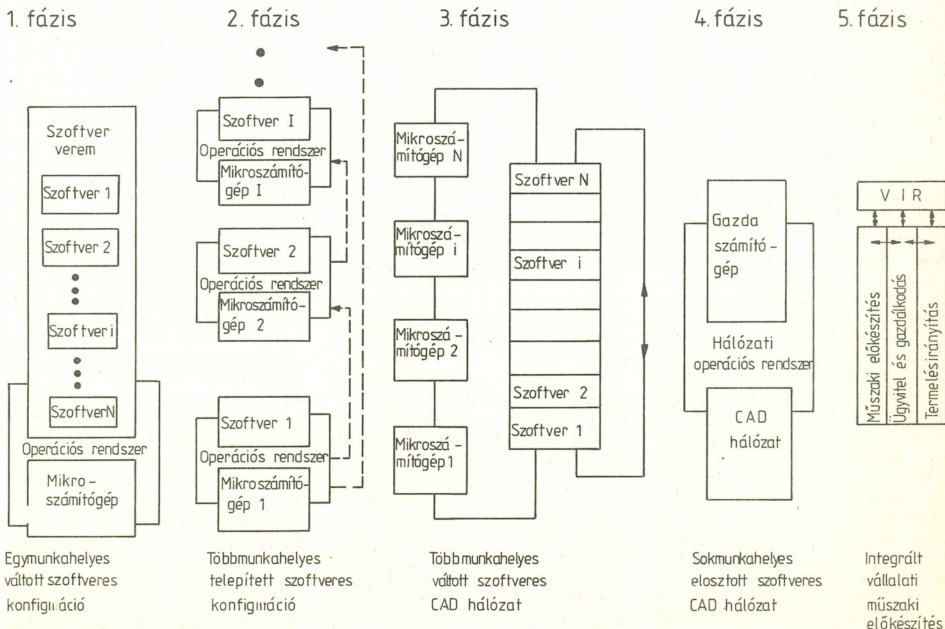
(Byte, 1991/szeptember)

Epilógus helyett
Mintagazdaság

A CAD alapjaiba bepillantó sorozat utolsó cikkéhez érkeztünk. Eddig tartott bolyongásunk ebben a különös világban, s mint jó tanítványokhoz illik, elérkezett az ideje annak, hogy visszapillantsunk: hogyan lehet hasznosítani azt, ami ránk ragadt. Cikksorozatunk befejezéseként a szerző egy olyan mikroszámítógépes CAD-környezetet mutat be, amely mintegy gyakorlati megvalósulása a sorozatot indulásakor — az egykori Mikroszámítógép Magazin 1990. márciusi számában — felvázolt elvnek.

Utat mutatni másoknak (avagy összeraktuk, amink van...)

A Budapesti Műszaki Egyetem Gépszerkezettani Intézete egy, az Országos Műszaki Fejlesztési Bizottságtól kapott megbízás keretében 1986 második felében felmérést végzett a magyarországi gépipari középvállalatok körében. A felmérés feltárta, hogy ebben a vállalati körben is egyre inkább kibontakozik az igény olyan alacsony költséggel megvalósítható, számítógéppel segített tervezőrendszerekre, amelyek a tervezési folyamat több tevékenységelemének támogatást tudnak adni, emellett követni tudják a vállalat CA technológiáinak fejlődését. Ami sajnálatos, hogy az iparvállalatok túlnyomó többségénél napjainkban úgy intézik el a CAD-fejlesztés problémáját, hogy (sok esetben szubjektív) megítélés alapján telepítenek egy vagy néhány, az aktuális feladatok igényelte szoftvert, és azok egymástól független használatán keresztül kívánják a vállalat tervezési tevékenységének hatékonyságát javítani. Ezt azonban gyakran megakadályozza, hogy a rendszerek közötti adatfolyam megszakad. Ugyanis a különböző gyártóktól származó eltérő rendszerek nem igazán támogatják az együttdolgozást. Még tovább menve: felhasználói interfészeik eltérő volta, a futtatásukhoz



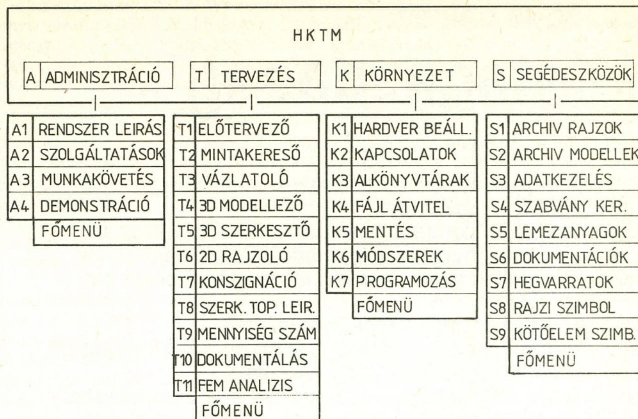
1. ábra. A vállalati CAD fejlesztés fázisai

szükséges hardver- és eszközmeghajtó beállítások, az adatok ismételt bevitele és főleg a más helyszínekre való átvonulás gyakran idegenkedést vált ki az alkalmazókban.

Közismert, hogy a munkaállomás-környezetekre közel tizenöt éve rendelkezésre állnak olyan szoftvercsomagok, amelyek a tervezési folyamat több tevékenységelemének támogatást tudnak adni. Ezek az eszközök azonban egyrészt szolgáltatásaikkal gyakran arányban nem álló kiadásokkal járnak, másrészt a kisvállalatok nem mindig tudják biztosítani optimális terhelésüket.

A fenti meggondolások tükrében, figyelembe véve a korábban említett felmérés tapasztalatait is, a BME GSZI célul tűzte ki egy, a gépészeti szerkezet- és részlettervezés egészét lefedő, egy munkahelyes, váltott szoftveres, adaptálható PC-CAD mintarendszer kidolgozását. A mintarendszer olyan tervezési környezet, amelynek szoftverelemei és adatbázisai lehetővé teszik az integrált CAD valamennyi tevékenységének bemutatását. A kifejlesztett tervezői környezet, jóllehet a bemutathatóság miatt konkrét feladatosztályhoz kötődik, kevés ráfordítással más objektumok tervezésére is átdolgozható. Ez elsődlegesen annak köszönhető, hogy a mintaobjektumként választott hegesztett keretszerkezet objektumtervezési módszertana és az alkalmazott rendszerfejlesztési elv nem korlátozza a rendszer más objektumok tervezésére történő átalakítását. A fejlesztés során messzemenően szem előtt tartottuk, hogy ne csak oktatási célú, hanem az ipari gyakorlat követelményeit kielégítő mintarendszer jöjjön létre. Vagyis a kifejlesztett rendszer rendeltetése szerint egyrész a CAD lehetőségeinek bemutatását, másrészt a hasonló fejlesztések végrehajtásához szükséges speciális szakmai ismeretek átadását hivatott szolgálni. A fejlesztési stratégiában jelentős hangsúlyt kapott, hogy a legnagyobb lehetőséget biztosítsuk kereskedelmi forgalmazású részrendszerek felhasználására.

A CAD mintarendszert önmagában és környezetében fejlődő, integrált erőforrás-együttesként értelmeztük. Feltételezésünk szerint, ha valamely, a CAD terén előzményekkel nem rendelkező magyarországi középvállalat ilyen irányú fejlesztést irányoz elő, akkor azt nagy valószínűséggel mikroszámítógép-beszerzésre alapozza. Kiindulásképpen egy, vagy kevés számú számítógépegység kerül be a napi használatba. Azért, hogy ezek a számítógépek optimálisan hasznosíthatók (azaz kihasználhatók) legyenek, célszerű egy munkahelyes, váltott szoftveres CAD-környezetként munkába állítani őket. Ez azt jelenti, hogy az egyes számítógépekre az összes olyan szoftvert rá kell telepíteni, amelyek az integrált CAD megvalósításához szükségesek. Mivel egyidejűleg csak egyetlen szoftver használatára van lehetőség, meg kell oldani, hogy a tervezők a lehető legegyszerűbb módon válthassanak a szoftverek között. Ennek érdekében pedig meg kell szervezni az egységesített adatforgalmat az egyes feldolgozó szoftverek között. A munkahely hardverkonfigurációját úgy kell kialakítani, hogy az a valamennyi szoftver működtetéséhez szükséges beviteli és kimeneti periferiális eszközt tartalmazza.



2. ábra. A mintarendszer menüstruktúrája

Felfogásunk alapján a váltott szoftveres munkahely lényegében egy ötfázisú átfogó vállalati műszaki számítástechnológiai fejlesztési projekt első időszakának célkitűzését jelenti (lásd az 1. ábrát).

A mintarendszer fejlesztése kapcsán másik súlyponti kérdésnek tekintettük, hogy a rendszer mutassa be a 3D geometriai modellezés alkalmazásából adódó újszerű tervezési módszertant, valamint a testmodellezés szerepét az adatalapú integrálás megvalósításában. A harmadik fontos szempont volt a szakadásmentes adatfolyamok megvalósítása, vagyis hogy a rendszer semleges formátumú kommunikációs fájlokkal (IGES és DXF) biztosítson adatátvitelt a különböző modulok között.

A mintarendszer hatáskörében (szolgáltatásaiban) a következő CAD részterületekre terjed ki: (a) szöveges dokumentációszerkesztés, (b) aktív és passzív rajzgenerálás, (c) komplex adatbázis-kezelés, (d) numerikus elemzés és szerkezetoptimálás, (e) mérnöki-mennyiség- és jellemzőmeghatározás, (f) technológiai előfeldolgozás, (g) szabványos adatkommunikáció és (h) részben a koncepcionális tervezés.

A rendszerben még nem került sor centralizált adatbázis létrehozására, hanem csak a funkcionális adatbázisszegmensek közötti adatállomány-cserének kommunikációs fájlokkal történő megvalósítására. Ez egyrészt a társított rendszerek integrálási lehetőségéből, másrészt az alapul választott számítógépes hardverkörnyezet lehetőségeiből adódik. Az ebből eredő korlátozás ellenére megvalósult az integrált CAD gyakorlatilag valamennyi eleme.

Szolid hardveren is működik

A keretszerkezet-tervező mintarendszer hardveralapjául kizárólagosan mikroszámítógépes eszközök szolgálnak. Az integrált CAD mintarendszer már matematikai társprocesszorral ellátott, 80286 processzoros AT-kompatibilis számítógépen is futtatható. Bővítőmemória és analóg beviteli eszközök (egér és tablet) a hatékony működés érdekében javasolhatók.

A GSZI-ben létrehozott konfigurációt felépítő eszközök tételesen a következők: az alapszámítógép egy IBM

PC/AT-kompatibilis mikroszámítógép, Intel 80286 processzorral. A számítógép utasításfeldolgozási sebessége 0,8-0,9 MIPS, szóhossza 16 bit, ütemfrekvenciája 8,47 MHz. Az Intel 80287 matematikai társprocesszor szóhossza 16 bit, ütemfrekvenciája 10 MHz. A mikroszámítógép belső program- és adattároló kapacitását a következők jellemzik: 640 kbájt alapmemória, 2560 kbájt bővítőmemória. A mikroszámítógép ASCII 108 karakteres billentyűzettel, 3 darab RS-232-c soros interfésszel, és 2 darab 8 bites Centronics párhuzamos interfésszel rendelkezik.

A grafikus megjelenítés céljára 1 darab multiszinkron színes megjelenítő ernyőt alkalmaztunk, amelynek felbontóképessége VGA üzemmódban: 800x600 képpont, EGA üzemmódban: 620x450 képpont. A palettaszínek száma: 64, az aktív opciószínek száma: 16. Mindezt egy AST — VGA gyártmányú meghajtókártya biztosítja. A rendszer második kommunikációs ernyője egy fekete-fehér alfanumerikus megjelenítő ernyő, amelynek karaktertérképe 80x25 karaktert foglal magában, vezérlését IBM monokróm kártya végzi. Papírmásoló eszköz három található a konfigurációban, mégpedig: 1 db Epson 1050 sornyomtató (LQ minőségű, 80 karakter/s nyomtatási sebességű), 1 db Epson GQ 3500 lézernyomtató (LQ minőségű, 30 lap/perc nyomtatási sebességű) és 1 db Hitachi gyártmányú asztali rajzgép. Ez utóbbi főbb jellemzői: A/3-as formátum, 0,3 m/s rajzolási sebesség, 0,5 mm beállási pontosság és 6 db toll váltott kezelésére képes.

A grafikaorientált beviteli eszközöket 1 db tablet (A/3-as formátum, 0,1 mm felbontási pontosság) és 1 db három funkciógombos GMOUSE egér képviseli. A háttér- és munkatárolási feladatokat 1 db merevlemezes gyorstárral (tárolókapacitása: 80 Mbájt, adathozzáférési sebessége: 0,27 ms) oldjuk meg. Átmeneti tárolási feladatokra 2 db floppy szolgál, amelyek tárolókapacitása: 1,21 Mbájt, adatelérési sebessége: 0,86 ms. Archív tárolási és biztonsági feladatok megoldására 1 db 60 Mbájt töltőkapacitású streamert építettünk a számítógépbe. Kiszolgálóeszközként kell megemlíteni az 1 db 2000 W teljesítményű szünetmentes tápforrást.

Az integrált CAD rendszer szolgáltatásai

A mintarendszer kereskedelmi forgalmazásban beszerezhető szoftvereket, illetve az integrált tervezést és adatkezelést támogató, saját fejlesztésű programokat egyaránt magában foglal. A rendszer menüvezérlésű, a funkcionális szoftverelemek eléréséről és a tervezést támogató környezet felállításáról (a környezet keretét adó) rendszertörzs gondoskodik. Ez a felhasználó számára barátságossá és könnyen kezelhetővé teszi a rendszert. Az integrált rendszer szoftverei négy blokkot alkotnak. Ezek rendre az adminisztrációs, a tervezőeszközök, a fájlkezelési és a tervezésisegédeszköz-biztosítási blokk (2. ábra). Az adminisztrációs blokk szoftverei a rendszer megismerésében, a kezelés elsajátításában, a rendszerhasználati statisztikák készítésében és a demonstrációk végrehajtásában nyújtanak támogatást. A mintarendszer a felhasználónak számítógéppel kezelhető formájú rendszerleírást biztosít a „kinél van a könyv?" szituációk elkerülésére. A felhasználó kulcsszavak kijelölése alapján kérhet információkat a rendszerről. A tervezőblokk szoftvereszközei az előtervezésben, a korábbi termékekre vonatkozó információk visszakeresésében, a grafikus vázlatolásban, a háromdimenziós test- és huzalváz-modellezésben, a műhely- és összeállítási rajzok készítésében, a gyártmányok strukturális leírásában, a gyártás-előkészítés végrehajtásában, a végeselemeken alapuló numerikus elemzések végrehajtásában, a szöveges és kiadvány jellegű dokumentációk megszerkesztésében nyújtanak támogatást.

A fájlszolgáltatási modul a rendszer telepítésében, a program- és adatfájlokat tartalmazó alkönyvtárak kijelölésében, fájlok rendszeren belüli mozgatásában, a rendszer streamerre mentésében, saját felhasználói módszerek programozásában, nyilvántartásában és futtatásában segítséget nyújtó programokat foglal magában. A negyedik blokk tervezést támogató segédeszközöket biztosít, így például archívrajz-, testmodell- és dokumentációtárolót, szöveges szabványkiolvasót, hierarchikus (grafikus és numerikus) adatbázis-kezelőt, IGES és DXF adatkonvertereket és rajzi szimbólumkönyvtárakat tartalmaz. A rendszerben kódolással összetartozóként nyilvántartva, de fizikailag elkülönítve tárolódnak az archív rajzok, modellek és írásos dokumentációk. A tárolás a termékek strukturális szintjeinek megfelelően történik.

A mintarendszer szoftverei MS-DOS operációs rendszer alatt futtathatók. Az alkalmazható legalacsonyabb változatszám a 3.1. A rendszer az integráltság elérésében rendkívül nagy mértékben alapoz az operációs rendszerre. Lényegében annak a kötegelt feldolgozási szolgáltatásaiból adódó lehetőségeit használja újszerű formában. Ezzel lehetőséget ad a DOS örökletes memóriakorlátozásainak és egyfeladat-irányultságának részbeni feloldására. A felhasználói környezet kialakításához az operációs rendszer mellett több segédprogram alkalmazása is szükséges.

A rendszerben a program-, az adat- és a szövegfájlok kezelése szabályozott. A rendszer telepítése a felhasználó által nem módosítható struktúrában történik. Ezt a rendszer váltott szoftveres és közvetlen állomány kommunikációs jellege teszi szükségessé. A feldolgozóprogramok az elérési utakat, a hozzáférési jogosultságokat, valamint a fájlok megnevezésének és kezelésének rendjét ismerik. A funkcionális modulok közötti szabályozott fájlátvitelt külön erre a célra kifejlesztett fájlkezelő szolgáltatás valósítja meg.

A rendelkezésre álló szoftverek váltott felhívását és futtatását a rendszertörzs végzi. Az operációs rendszertől átvéve a vezérlést a rendszertörzs meghívja a főmenüt. A felhasználó által végrehajtott kiválasztás eredményeként hozzáférhetővé válik a kívánt feldolgozóprogram. A rendszertörzs megengedi, hogy a feldolgozóprogramokból visszalépjünk a DOS-szintre, és ott operációsrendszer-parancsot vagy programfuttatást hajtsunk végre. Természetesen ez utóbbi csak akkor lehetséges, ha a számítógép megfelelő méretű közvetlen, virtuális vagy kiterjesztett memóriával rendelkezik. A rendszertörzs a feldolgozószoftverek számára a futtatási környezetet automatikusan előkészíti, azaz behívja a szükséges meghajtószoftvereket, felszabadítja az igényelt memóriaterületet és, ezt követően futtatja a szoftvert.

Ablak a világra

A mintarendszer ablakozó menükezelővel készült. A megfelelő menüopciók kijelölése után a rendszertörzs program felfüggeszti működését, és elindítja a szükséges programokat. A kiválasztott feladat befejezése után ismét a menü jelentkezik be. Ha a felhasználó a menü kilépés funkcióját választja, akkor a rendszertörzs egyrészt aktivizálja az archiváló programot, másrészt meghívja az adminisztrációs modult. Az archiváló program automatikusan működik, beírja a naplófájlba a kilépés időpontját, és befejezi működését. A felhasználó a beépített Turbo Pascal 6.0 programozási szoftverrel saját célprogramokat fejleszthet, és ezeket módszerbázisban tárolhatja.

Horváth Imre

A szoftver jogvédelméről

A Neumann János Számítógéptudományi Társaság több más érdekelt szervezettel együttműködve október közepén rendezte meg nemzetközi konferenciáját a szoftver jogvédelemről.

Elgondolkodtató, hogy a jelentős hirdetési és szervezési erőfeszítések — lásd például Alaplap 91/9 — ellenére mindössze hetven regisztrált részvevő jött össze. Bár a szoftver védelme — avagy védtelensége —, a kalózkodás a fejlesztők és felhasználók állandóan felmerülő gondja, a többség még mindig nem érzi elengedhetetlennek, hogy a kérdéskörben alaposabban tájékozódjék.

Így a konferenciából inkább workshop lett, annak viszont jól sikerült. Találkoztak a szoftvervédelem jelentős európai művelői, továbbá egy amerikai és egy dél-afrikai előadó is részt vett a tanácskozásban. Húsz külföldi és hat magyar szakértő tartott előadást. A saját országaikat jellemző körülmények bemutatása mellett szinte mindenki érintette a szoftvervédelem aktuális kérdéseit.

— A szoftver intézményes jogvédelmének lehetőségei. A több évtizedes vita itt is kialakult: vajon a szabadalom jó-e, milyen előnyei vannak egy sui generis, külön erre a célra kialakított rendszernek. Jelenleg a világ döntő részében a szoftver szerzői jogi védelmét részesítik előnyben.

— Az Európai Közösség irányelve és az egyes országok jogrendszerének harmonizálási lehetőségei.

— A szerzők személyhez fűződő és vagyoni jogainak kezelése.

— A munkavállaló szerzők jogai, a munkaadó rendelkezési foka és joga az alkotás felett.

— Az eredetiség kritériumai, értelmezése.

— A szoftvervédelmi idő (ami általában 50 év a szerző halála után). Ez egy szoftver életciklusával összehasonlítva kissé hosszúnak tűnhet.

— A reverse engineering, a visszafejtés sajátosságai. Jelenleg a magyar jog szerint csak a szerző hozzájárulásával végezhető, sok országban ez anélkül is megengedett, ha más szoftverrel történő összekapcsolás, együttes működés céljából fejtik vissza az interfész részt.

— A szoftver más programokkal való közös használatával is jelentősen összefügg a kizárólagos jogok birtoklása és a tisztességtelen verseny elleni törvények monopolhelyzetet támadó passzusai közötti ellentmondások tisztázása.

— A megvásárolt program használatának mélysége (a futtatás mellett mi jogszerű). Általában megengedett még a szoftver tanulmányozása, a funkciók vizsgálata, de a program kereskedelmi célú reprodukálása természetesen már nem.

— A kalózkodás okai, az illegális másolás elleni küzdelem lehetőségei, a jogsértések különböző szankciói (büntetőjogi következmények; szoftver- és hardverkulcsok; felhasználóbarát marketing módszerek).

— A szoftverpiac sajátosságai, a licencszerződések tartalma, fajtái.

— A szoftverdokumentálás, a prioritás védelmi hatása és a regisztrálás lehetőségei.

Szóba került még néhány érdekes bírósági eset külföldi példákból, a magyar személyi jövedelemadózás sajátos és ellentmondásos helyzete, valamint egy figyelemre méltó megközelítés, melynek szerzője feleslegesnek tartotta a mai — szerinte túl komplikált — jogi védelmi rendszert. Érvelésének lényege: ha a gyakorlatban úgyis elterjedtebb az illegális használat, jobb lenne ezt tudomásul venni, mivel a jogvédelem akadályozza a szoftver terjedését, használatát, így a fejlődés és haladás ellensége. Ezen nézet szerint csak a különlegesen nagyértékű, egyedi fejlesztésű komplex rendszereket kellene jogi védelemben részesíteni. (Lehet, hogy mi Magyarországon már itt tartunk?... Természetesen ilyen megoldásról szó sem lehet, ez nagymértékben sértené a fejlesztők érdekeit.)

Ahogyan a számítástechnikában is állandó a mozgás, a számítástechnikai programok védelmének kérdéskörében is szakadatlan a fejlődés. A WIPO (Szellemi Tulajdon Világszervezete) szakértői bizottsága most készíti a Berni Uniós Egyezmény kiegészítő jegyzőkönyvét, melyben a szoftverjogvédelem modelljét kínálják az Egyezmény tagországainak és a jövőben csatlakozóknak. A szakértők állandó tapasztalatgyűjtése és konzultációi nyomán érlelődik ez az anyag. Ficsor Mihály, a WIPO Szerzői Jogi Osztályának igazgatója — a konferencia legmagasabb rangú előadója — záróértékelésében kifejtette, hogy Budapesten a referátumokban és a viták során több hasznos vélemény is elhangzott, melyeket figyelembe fognak venni novemberben a WIPO következő szakértői összejövetelén.

Remélhetőleg Magyarország — hiszen úttörő szerepet játszott a szoftver védelmére vonatkozó jogalkotásban — a jog alkalmazásában sem fog teljesen a perifériára szorulni. A fekete és fehér szoftverforgalmazás közötti minden árnyalat jelenlétével büszkélkedő magyar piacon illenék ismerni a játékszabályokat.

(Az előadások szerkesztett kötete még kapható a Neumann Társaság titkárságán: Budapest V., Báthori u. 16.).

O. G.

Csatlakozás a 87-es buszhoz

Mr. Intel Londonban

Ha a kaliforniai turista (aki esetleg az Intelnél dolgozik) a londoni földalatti valamelyik állomásáról el akar jutni a 87-es buszhoz, bedob két egyfontos érmét az automatába, megkapja jegyét és a visszajáró aprópénzt. Genfben két egyfrankosért megkapja az 1,80-as jegyet, az apró elvész. Budapesten a két tízforintosért semmit sem kap, hiszen a 20 nem osztható 12-vel. Érthető az automaták eltérő viselkedése: mindegyiknek más a processzora. Még ennél is meglepőbb az ugyanazt az Intel 80x87 matematikai processzort használó programok viselkedése. A cikkhez kapcsolódó ábrák és táblázatok terjedelmi okokból a mágneslemez mellékletre kerültek.

Az egy gépi szóban ábrázolható számok tartományát jelentősen kitágítja a lebegőpontos értelmezés. Ennek „klasszikus" formája:

$$s = e * m * 2^k \quad (1)$$

ahol:

s az ábrázolt szám értéke
e az ábrázolt szám előjele („0" bit: +1, „1" bit: -1)
m a szám mantisszája (1 < = m < 2)
k a szám karakterisztikája

Akkor, ha m nincs a megadott tartományban, s = 0. Ennek megfelelően létezik az ábrázolható legkisebb abszolút értékű szám, azaz abban az esetben, ha a számítás során m 1-nél kisebb lenne, s-et 0-val helyettesítik. Ezt nevezik alulcsordulásnak (underflow).

Hasonló megfontolások alapján 2 helyett más alapot is lehet alkalmazni, s ezzel összhangban kell m értékhatárait megállapítani. Ha például 16-os alapot választanánk, mint az IBM System/370-es gépeknél történt, a követelmény: 1 < = m < 16 lenne.

A normalizálás indoka: ha az m mantissza hossza változatlan, minden multiplikatív művelet ugyanannyi jegyre lesz pontos. (Decimálisan: lg(2)*b = 0.30103*b, ahol b a mantissza bitjeinek száma). Ha m nincs normalizálva, hossza úgy csökken, ahogy a bal oldalon szaporodnak a nullák, s ezzel együtt romlik az eredmények pontossága. A programozó megkövetelheti, hogy bízhasson adataiban, aminek egyik biztosítéka lehet a nem normalizált számok 0-kénti értelmezése.

Mindig a feladattól függ, hogy mekkora az a számítási részeredmény, amelyet már 0-val lehet helyettesíteni. Az alulcsordulás a jól tervezett gépeknél programmegszakítást okoz, és a programozó hatáskörébe tartozik a szükséges intézkedések megtétele. Például az IBM System/370 OS operációs rendszer ERRSET rutinjának meghívásával elő lehet írni, hány alulcsordulási esemény után állítsa le a rendszer a programot, ezen belül pedig hányszor írjon ki hibaüzenetet, amely tartalmazza a megszakítást kiváltó utasítás helyét is.

Az Intel 80x87 matematikai processzor az (1) formula szerint dolgozik. 8 lebegőpontos regisztere 80 bit szélességű, ebből m 64 bites (kb. 19 decimális jegy pontosság), k pedig lehetővé teszi a kb.

$$3.4*10^{-4932} < abs(s) < 1.2 * 10^{4932}$$

közötti számtartomány használatát.

A 80 bitnek megfelelő 10 bájt hosszú információ nem illeszkedik a bájtszervezésű gépeknél (például IBM System/360, 370) történetileg kialakult 4, illetve 8 bájtos szimpla és dupla pontosságú számábrázoláshoz, amit az IBM PC-k és klónjaik is alkalmaznak. A 80x87-es matematikai processzor, ha a 8086-os CPU egy WAIT utasítást hajt végre, az adatbuszról megkapja a programkörnyezetből kideríthetően 4 vagy 8 bájt hosszú szimpla vagy dupla gépi szó tartalmát, s végrehajtja az előírt műveletet. m és k ennek a 4 vagy 8 bájtos gépi szó bizonyos bitjeinek so-

Az Intel 80x87 processzor hibajelzései

Nem normalizált operandus. A hibajelzés arra utal, hogy vagy a matematikai művelet valamelyik operandusa, vagy eredménye nem felel meg a választott szóhossz m-re vonatkozó korlátainak. Letiltott megszakítás esetén a további számítások a pontatlanná vált értékekkel folytatódnak.
Alulcsordulás. Az érték túl kicsi a választott szóhosszhoz tartozó ábrázolási formához. Letiltás esetén: számolás nem normalizált operandussal vagy 0-val helyettesítés az értéktől függően.
Túlcsordulás. Az érték túl nagy a választott szóhosszhoz tartozó ábrázolási formához. Letiltás esetén: gépi végtelennel helyettesítődik.
0-val osztás. Letiltás esetén: az eredmény „NaN" (Not a Number). Gyakori megoldás a hibajelzés olyan kezelése, hogy az eredményt gépi végtelennel helyettesítik.
Kerekítés. A 80 bites ábrázolásról a 32 vagy 64 bitesre konvertálás során kerekíteni kellett. Ennek a hibajelzésnek a kezelése igen sokféle lehet, általában nincs dokumentálva.
Érvénytelen művelet. Kísérlet NaN-nal vagy gépi végtelennel való számításra; a 8087-es vermének túlcsordulása. (Az utóbbit csak hibás fordítóprogrammal lehet elérni. Ezt az esetet felesleges felhasználói szinten elemezni.)
A hibajelzés letiltása esetén a NaN, illetve a gépi végtelen öröklődik.

rozatából képzett egész (integer) számok. Figyelemre méltó, hogy a kétféle szóhosszúság egyben eltérő m- és k-hosszúságot is jelent (lásd az 1. és 2. ábrán).

Az egyszerűség kedvéért k helyett a K_4, illetve a K_8, és m helyett az M értékeket tárolják, melyek értelmezése:

$k = K\ 127$ (4 bájt), (2a)

$k = K\ 1023$ (8 bájt). (2b)

Ha $K_i > 0$, akkor:

$m = M + 1$, (3a)

egyébként:

$m = M$, (3b)

ahol $0 <= M < 1$ (azaz csak m törtrészét tárolják).

Az Intel 80x87 hibajelzést ad, ha a műveletek valamelyik operandusa vagy eredménye nem felel meg a számábrázolási szabályoknak. A hiba kezelése a program feladata. A lehetséges hibajelzések mindegyike letiltható. Nyilvánvaló, hogy a letiltások kombinálása csak a programozó elhatározásától („filozófiájától") függ. Ennek az a következménye, hogy a magas szintű nyelveken írt felhasználói programok különböző eredetű fordítóprogramokkal létrehozott, futtatható (.EXE) változatai eltérő eredményeket hozhatnak létre.

A processzor működésmódjából következik, hogy letiltott hibajelzések esetén a nem normalizált számokkal végzett műveletek olykor (elsősorban additív műveletek során) NaN-t generálnak. Ez természetesen a program logikai, de nem fizikai halálát jelenti: a gép dolgozik, esetleg nem konvergáló iterációval tölti idejét. (Mondanunk sem kell, hogy Murphy tanítása szerint ez sohasem a tesztadatokkal történik, hanem a program használatbavétele után pár hónappal.) Egy ilyen eredetű programhibát bogarászva nagyon egyszerű algoritmussal is sikerült vallatóra fogni (abécé-rendben:) BASIC, FORTRAN-77, PASCAL és SIMULA rendszereket. A programok rendre az 1-4. listákon láthatók. (A SIMULA rendszerből csak az ALGOL-60 részhalmazra volt szükség.)

1. lista

```
10 A=1
20 A=A/3 : B=A
30 A=A+A : PRINT A
40 IF A=B THEN END
50 GOTO 20
```

2. lista

```
PROGRAM BUSZ87
CDOUBLE PRECISION A
EQUIVALENCE (A,IA)
A=1.0
20 A=A/3.0
B=A
A=A+A
WRITE (*,*) A,IA
IF (A.EQ.B) STOP
CIF (IA.LE.1) STOP
CIF (IA.EQ.0) STOP
GO TO 20
END
```

3. lista

```
PROGRAM busz87;
VAR a,b: REAL;
BEGIN
a:=1;
b:=0;
WHILE ab DO BEGIN
a:=a/3;
b:=a;
a:=a+a;
WRITE (a);
WRITELN (' ');
END;
END.
```

4. lista

```
SIMULATION
begin
real A,B;
A:=1;
L: A:=A/3;
B:=A;
A:=A+A;
OUTREAL(A,7,13);
OUTIMAGE;
if A # B then go to L;
end;
```

A vizsgálat célja volt megtudni, hogy a különböző rendszerek hogyan kezelik az alulcsordulást osztás esetén. Mivel a nem normalizált összeadással könynyen lehet baj, a másféllel való osztást a 3-mal osztás és 2-vel szorzás helyett az osztás és a hányados önmagával való összeadása helyettesíti. Alulcsordulás esetén a szám 0 lesz, tehát a = a + a várható. Ez lehet az iteráció befejezésének feltétele. Nos, a feltétel néhány esetben a-nak nem nulla értékénél is teljesült! Az 1. táblázat szintén abécérendben BASIC-eredményeket mutat. Mivel a programok BASIC értelmezőrendszerekkel futottak, a háromféle befejeződés a három fejlesztő cég eltérő logikáját is mutatja. A Borland-féle Turbo BASIC (erre a FORTRAN-eredményekből lehet következtetni) nem bírálja felül a 80x87-es viselkedését. K eléri a 0-t; m legkisebb értéke pedig egy 1-es bitnek látszik a második legkisebb helyi értéken.

Mivel a 80x87-es belső ábrázolásában a már nem látható helyi értékeken is lehet 1-es, a kerekítési hibák miatt az

A változó a további műveletek során ugyanezt az értéket kapja vissza: a felhasználói program úgy dolgozik, mintha

$2.803*10^{-45}$

értéke valóban nem változna a másféllel osztás során. A Microsoft rendszerek a „klasszikus" értelmezést szimulálják. Korrektnek a Summit cég megoldása látszik: leállás hibajelzéssel.

A FORTRAN-77 nyelv EQUIVALENCE utasítása lehetőséget ad ugyanazon gépi szó bitkombinációjának többféleképpen történő értelmezésére. A lebegőpontos változók az 1. és 2. ábrán bemutatott bittérképét könnyebb visszafejteni, ha azt decimális egészként használjuk. A FORTRAN-mintaprogram 2. táblázatbeli eredményeiből a három programozócsapat eltérő ideológiája kitűnik. A Lahey FORTRAN fordítóprogram m legkisebb, a Microsoft m második legkisebb helyi értékén lévő 1-es bit esetén nem módosítja tovább a példa számsorát. A Ryan-McFarland cég egy fél bájtot figyel: a 15-ös érték 4 darab 1-es bitet jelent. Ha a 2. listán látható programban megjegyzésként szereplő másik leállási feltételeket használjuk, a Lahey és a Ryan-McFarland egyformán viselkedik: ha m 1-ese eléri a legkisebb helyi értéket, az az ábrázolható legkisebb számot jelenti. A Microsoft ugyanezt a második legkisebb helyi értéken teszi.

A PASCAL egyetlen példája szerint a Borland 8 bájtos ábrázolással az m második legkisebb helyiértékén lévő 1-est már nem tudja kezelni. A SIMULA nyelv első PC-re írt fordítóprogramjának (Gáspár András munkája) szabad terjesztésű demóváltozata a real és a long real változókat nem különbözteti meg: a 4 bájtos alulcsordulásnál hibajelzéssel leáll.

A zérussal osztás, a túlcsordulás, továbbá a definiálatlan 0/0 műveletek „eredményeit" a 4. táblázat tartalmazza. A különböző megközelítések itt is jól láthatóak. A 0-val osztás előfordulhat többek között úgy is, hogy a nevező egy előző alulcsordulás miatt lett zérus. Ebben az esetben jó megoldás lehet az eredmény gépi 8-cal való helyettesítése. Akkor, ha az előélet más, más eljárás lehet helyes. A programozó felelőssége a jó algoritmus helyes kódolása.

A gyakorlatban előforduló problémák jelentős részét megoldja, ha 8 bájtos változókat használunk, de mint a fentiekből látszik, ez sem csodaszer. Kár.

Szondi Egon János

A Pascal Saga — a rokonok

… a manókirály és a grófnő

Az előző hónapban közöltük e „családregény" első felét; most — ígéretünkhöz híven — a még hátramaradt részeket tárjuk olvasóink elé. Azért különlegesen is érdekes ez, mert a Pascal-bővítések közül néhányan önálló életre keltek…

Ilyen például a Brinch Hansen-féle Concurrent Pascal, amely multitaszkos bővítéseket tartalmazott, valamint az Object Pascal, amelyet az objektumorientált programozáshoz szükséges eszközökkel láttak el. Azonban ezek a nyelvek részben speciális alkalmazási területeik, részben a különböző Pascal-implementációkban is meglévő, hasonló bővítések miatt nem terjedtek el széles körben.

Érdekes módon Wirth nagyon határozottan elzárkózott mind a Pascal egységes bővítése, mind a szabványosítási kísérletek elől. A mester már régen túltette magát a Pascalon, és a real-time rendszerek, valamint a hardvereszközök programozásának problematikájával foglalkozott. Erre a célra — miért is ne? — létrehozott egy nyelvet, amelyet a Pascalból származtatott, és alapvető struktúrájára való tekintettel Modulának nevezett el.

Modula 1, 2, 3

A Modula letisztult Pascal-szintaktikával rendelkezett, és a Pascal nyelvi elemeinek csak egy kis részét foglalta magában: nem volt általános célú programozási nyelv, ezért hiányoztak belőle a Pascal fájl-, halmaz- és mutatóstruktúrái. Noha a programozás nehézkessé vált a Pascalban megszokott adatstruktúrák nélkül, a modulok olyan jelentős fejlődést jelentettek a Pascal blokkstruktúrájához képest, mint a Pascal blokkstruktúrája a Fortran szubrutinhívásaihoz képest.

Bár a Modulát még ma is használják néhány helyen, Wirth megszakította a vele kapcsoltos kutatásait is, miután egy évet eltöltött a Xerox Palo Alto Research Centerben (PARC). Itt Wirthnek lehetősége volt megismerkedni az Alto számítógéppel, és a Pascal egyik utódjával, a Mesa programozási nyelvvel. Svájcba visszatérve Wirth olyan munkába kezdett, melynek célja egy, az Altóhoz hasonló, de szerényebb számítógéprendszer létrehozása volt. Mivel az egyik feltétel értelmében a később Lilithnek elkeresztelt személyi munkaállomás minden programját — az eszközmeghajtóktól az operációs rendszeren át az alkalmazói programokig — egyetlen programozási nyelven kellett megírni, Wirthék elővették a Pascalt és a Modulát, és alaposan átdolgozva, átgyúrva őket, egy új nyelvet alkottak: a Modula-2-t.

A Modula-2 általános célú rendszerimplementációs programozási nyelv, amely magában foglalja elődei legjobb elemeit. A Modula-2 utasításkészlete és adattípusai a Pascaléinak kibővítései; a letisztult, rendezett szintaktika, a modularitási elv, a gépközeli programozás és a multiprocesszálás a Modulában található eszközök javított változatai. Ezenfelül a nyelv szisztematikusan kiküszöbölte a Pascal hiányosságait és problémáit. A Modula-2 népszerűsége mind az akadémiai, mind az ipari felhasználók körében rohamosan nő az utóbbi időkben. A felsőoktatási intézmények nagy részében a Modula felváltotta a Pascalt mint a számítástechnikai oktatás alapnyelvét. Mivel a Modula verifikálható, rendelkezik gépközeli programozáshoz és real-time alkalmazásokhoz szükséges eszközökkel, ezért az iparban is egyre szívesebben alkalmazzák. A modulstruktúra miatt előszeretettel használják nagy rendszerek létrehozására.

A Modula-3-at a DEC és az Olivetti kutatói tervezték. A Modula-3 a szintén a DEC-nél készített, kibővített Modula-2-nek, a Modula-2+ nyelvnek a továbbfejlesztett változata. A Modula-2+-ra egyébként a Modula-2-n kívül a PARC Cedar nyelve gyakorolt még hatást; ez utóbbi viszont a Mesa utódja

Családfa

Niklaus Wirth
ETH
1968 — Pascal
1979 — Modula-2
1988 — Oberon

1984 — Object Pascal

Xerox PARC
1977 — Mesa
1982 — Cedar
1980 — Euclid

DEC SRC
1984 — Modula-2+
1988 — Modula-3

DoD
1983 — Ada
(ANSI/MIL STD 1815A)

Kezdetben vala a Pascal, amelynek egyik implementációja a Mesa, ennek hatására tervezte Wirth a Modula-2-t. A Modula-2 és a Cedar tanulságainak inspirációjára a DEC kutatási központjában létrehozták a Modula-2+, majd ebből a Modula-3 programozási nyelvet. Az Object Pascal a Pascal legrégebbi, objektumorientált kiterjesztéseket tartalmazó implementációja. Az Ada az Amerikai Védelmi Minisztérium által létrehozott Pascal alapú programozási nyelv. Az Oberon a Modula-2 utóda.

— a Mesa azonban, mint láttuk, egy Pascal-változat —, amely nyelvnek a Modula-2 létrejöttét is köszönhetjük. (Mindenki mindenkivel, micsoda család!) A Modula-3 mint a Modula-2+ letisztult változata 1989-ben lett kész, és Wirth beleegyezésével kapta a nevét.

Az új nyelvként a DEC és az Olivetti fejlesztői a Modula-2-nél hatékonyabb, de ugyanakkor biztonságosabb rendszerimplementációs nyelvet akartak alkotni. A Modula-3 típusrendszere eltér a Modula-2-étől, és a futásidejű hibákkal szemben is nagyobb biztonságot nyújt. A nyelv bővítéseihez tartozik még a „garbage collection" (a dinamikus allokációs terület, az ún. heap fragmentálását és a fel nem szabadított memóriaterületek visszanyerését végző rendszer), a kivételkezelés, a threadek (könnyű processzek) és az objektumorientált programozás támogatása.

Az Oberon

Aki azt hiszi, hogy Wirth professzor elégedetten csücsült a babérjain a Modula-2 megalkotása után, nagyon téved. Alig fejeződött be a Lilith-projekt, az ETH-n beindították a CERES (Computing Engine for Research, Engineering and Science) programot. A CERES egyfelhasználós hálózati munkaállomás, amelynek teljes rendszerét Oberon nyelven írták. Az Oberon — nevét egyébként az Uránusz egyik holdjáról kapta, amelyet viszont a Szentivánéji álom manókirályáról neveztek el — legmeglepőbb tulajdonsága az, hogy a mai masztodon szoftverekkel teli világban az ellenkező utat járja: elődjénél lényegesen kisebb és egyszerűbb.

A CERES programot 1985-ben még egy új processzortípuson alapuló, színes Lilith-munkaállomás létrehozására indították. A rendszerimplementációs nyelv a Modula-2 lett volna, de mivel Wirth szerint a Modula-2 nem rendelkezett megfelelő eszközökkel a felhasználói/programozói bővítésekhez, ezért a típusbővítés lehetőségével ellátott Modula-2-ből egy új nyelv, az Oberon született. Ha már a sebészkés dolgozni kezdett, Wirth egyéb — a projekt számára felesleges — nyelvi elemektől is megszabadította a Modula-2-t, mivel az új rendszernek és az Oberonnak a lehető legegyszerűbbnek kellett lennie.

Eltűntek a variáns rekordok, a felsorolt és intervallumtípusok, a múlté a FOR ciklus és a gépszintű tulajdonságok. Sok más nyelvi elemet áramvonalasítottak: a tömböket, a mutatókat, a modulkönyvtárakat, és még jó pár dolgot.

Persze azért bővítések is akadnak. Ezek közül a leglényegesebbek az Oberon OOP-szerű bővítései, a típuskiterjesztés, és az ezzel kapcsolatos nyelvi elemek. Újdonság továbbá még a tetszőleges dimenziójú, nyitott tömbparaméter, a „garbage collection", és a Modula-2-beli ún. áttetsző típusok általánosítása.

Az Oberon nemcsak egyszerűen egy programozási nyelv, hanem komplett operációs rendszer is a 32 bites hálózatba kapcsolt CERES munkaállomások számára. Az Oberon rendszer egyik érdekes tulajdonsága, hogy a rendszer maga is tetszőlegesen bővíthető a programozó által. A modulkönyvtárak a rendszer részét képezik, és minden futó program rendelkezésére állnak. Ezenfelül a könyvtárakban található, paraméter nélküli eljárások az operációs rendszerből parancsként is elindíthatók.

Az Oberon rendszer sok olyan elemet tartalmaz, amelyek véleményem szerint meghatározóak lesznek az eljövendő számítógép- és operációsrendszer-generációk számára. (Itt hívjuk fel a figyelmét olvasóinknak, akik esetleg többet szeretnének megtudni az Oberonról: a Modula-2 sorozat cikkeiben még részletesen lesz szó a CERES-ről és az Oberonról. — A szerk.)

A költő lánya után: Ada

A Pascal család egy tekintélyes tagja — a távoli rokon minőségében — az Ada programozási nyelv. Az Adát az Amerikai Védelmi Minisztérium (DoD) kezdeményezésére és igényeinek kielégítésére hozták létre 1983-ban. (Pontosabban ekkorra fejezték be a nyelv tervezését.)

A nyelv a világ első programozójáról, Ada Augusta Lovelace grófnőről (a költő Byron lányáról) kapta a nevét. Lovelace grófnő Charles Babbage mechanikus számítógépe, az analytical engine számára konstruált programokat egy olyan korban, amikor számítástechnika még nem is létezett.

Az Ada tulajdonképpen egy amerikai katonai szabvány, az ANSI/MIL STD 1815A 1983, ahol 1815 Lovelace grófnő születési éve. Ezenfelül az Ada szó az amerikai kormány által bejegyzett védjegy. Az Ada mint a Pascalon alapuló programozási nyelv igyekezett magában egyesíteni a már ismert programozási nyelvekben található összes megoldást, ötvözve ezeket a megrendelők által felállított preferenciákkal. Mondani sem kell, hogy a végeredmény meglehetősen kétséges. Az Ada bonyolult, sok felesleges funkciót tartalmazó nyelv, amelynek implementálása szinte lehetetlen feladat elé állítja a fordítókészítőket, ráadásul jelenlegi formájában nem is verifikálható, ami az ipari, katonai és űrkutatási feladatokban kockázatossá teszi a használatát. (A verifikálhatóság az a tulajdonság, hogy egy adott nyelven írt program viselkedését tetszőleges peremfeltételek mellett, tisztán a forráskód ellenőrzésével meg lehet határozni — anélkül, hogy a programot le kellene futtatni.)

Mi értelme van akkor ezzel a nyelvvel küszködni? Ennek két oka is van. Egyrészt: aki a DoD által pénzelt, zsíros haszonnal járó fejlesztésekben részt kíván venni, annak az elkészített programokat Ada-forrásokkal kell dokumentálnia. Másrészt: természetesen az Ada azért nem teljesen alkalmatlan programozási feladatok megoldására. Moduláris felépítése miatt — a Modula-2-höz hasonlóan — a több programozó által készített nagy rendszerek programozását megkönnyíti. Az Adában már gigantikus programok is készültek, tehát mégiscsak jó valamire, de mivel a használata szinte kikényszerített, nagyon sokan ódzkodnak tőle. (Ezt a jelenséget mi itt Kelet-Európában nagyon is jól ismerjük.)

A mágneslemezen található a 91/10. szám lemezmellékletén található két, C nyelven írt program Turbo Pascal nyelvű megfelelője. Ezek és más programok a sorozatot lezáró, a fordítókat összehasonlító cikkhez készültek. A teszteredmények és a teszt objektivitásának ellenőrzését megkönnyítendő, a tesztekben szereplő összes program forrása megtalálható lesz majd a lemezmellékleten.

Villányi László

Gyöngyszemek (utilityk) és mag

A Unix-kagylón belül és kívül

Sorozatunk előző két részében általános Unix-jellemzőkkel ismerkedtünk. Térjünk most át az operációs rendszer felépítésének részletesebb ismertetésére.

A Unix, mint azt már többször említettük, egy többfelhasználós multitaszkos operációs rendszer. Többfelhasználós — mert terminálokon keresztül egyszerre több ember is használhatja a számítógépet — és multitaszkos: mert egy felhasználó egyidejűleg több programot is futtathat. Az előnyök közismertek: például amíg egy program fordul, editálhatunk egy másikat, vagy levelezhetünk más felhasználókkal stb.

A Unix operációs rendszer általános szerkezetéből látható, hogy a Unix mint rendszerprogram különböző szintű, egymásra épülő szolgáltatások összessége. A Unixnak nevezett rendszert a következő elemek alkotják:

— A kernel, a Unix magja, amely a hagyományos operációsrendszer-feladatokat (programütemezés, erőforrásgazdálkodás stb.) végzi.

— Jól megválasztott segédprogramok sokasága, amelyek kombinációjával bonyolultabb feladatok is elvégezhetők.

— Hierarchikus fájlrendszer, amely egységesen kezeli a hagyományos értelemben vett fájlokat és a perifériákat.

— A parancsértelmező, a shell, amely kapcsolatot teremt a felhasználó és a mag, illetve a segédprogramok között (amelyen keresztül a segédprogramok közvetlenül is használhatók).

A kernel

A kernel (mag) program, ahogy az a nevében is benne van, az operációs rendszer lelke. Az alapfeladatokat látja el, azaz ütemezi a programokat, gazdálkodik a rendelkezésre álló memóriával, kezeli a különböző hardvereszközöket, perifériákat stb. Az összes többi operációsrendszer-feladatot a segédprogramok (utility programok) végzik.

Egy multitaszk rendszerben a programok a központi egység (CPU) idejéből csak szeleteket kapnak. Ilyenkor meghatározott ideig futhatnak, majd át kell adni a helyüket egy másik taszknak. Ha ez a helycsere elég gyakran megtörténik, az lesz az érzésünk, hogy programjaink „egyszerre" futnak.

Helycsere általában valamilyen esemény hatására történik. Ilyen események a következők lehetnek:

— Az éppen futó program valamilyen I/O műveletre vár. Például, nyomtatás esetén meg kell várnia, amíg az adatok kikerülnek a nyomtatóra. A várakozás ideje alatt fölöslegesen foglalná a gépet, ezért várakozó állapotba kerül és valaki más indul el helyette.

— A várakozó programok közül valamelyiknél bekövetkezik az az esemény, amire várnia kellett, például véget ért a nyomtatás. Ilyenkor a program a futásra várakozók közé kerül, és ha bizonyos feltételeknek megfelel, el is indul.

— Lejár az az idő, amit az adott taszk megkapott. Ennek akkor van jelentősége, ha csak futásra váró programok vannak, azaz az előző két esemény közül egyik sem történik meg. Ahhoz, hogy ilyen esetben is mindenki szóhoz jusson, meghatározott időnként lezajlik a taszkváltás.

Minden programnak van egy ún. prioritása. Ez egy mérőszám, amelynek alapján a programok között fontossági sorrend állítható fel. A kernel fő feladata az, hogy a prioritásokat figyelembe véve lebonyolítsa a taszkok közötti helycserét. A futásra váró programok közül kiválasztja a legnagyobb prioritásút, és azt futtatja, azaz annak engedi át a gépet.

A segédprogramok

A Unix alapvető filozófiája az, hogy az operációs rendszer magja, a kernel, csak az alacsonyabb szintű feladatokat végzi el. Minden egyéb szolgáltatást a kernelre épülő segédprogramok biztosítanak. Ez az elv lehetővé teszi, hogy a Unix állandóan bővüljön, de megvalósítható az is, hogy a segédprogramok az alkalmazásoknak megfelelően cserélődjenek. Ez nagy rugalmasságot biztosít a Unix rendszernek.

Ennek a felépítésnek köszönhető, hogy a Unix könnyen átvihető más számítógépekre is. A Unixot alkotó programoknak csak kb. a 10%-a gépfüggő bináris kód, ezért csak ezt a részt kell a különböző gépekre újraírni. A C nyelven íródott maradék, amelynek legnagyobb részét a segédprogramok alkotják, minden további nélkül átvihető.

A Unix portabilitása a felhasználóknak és a fejlesztőknek is előny, mert géptől függetlenül ugyanabban a környezetben dolgozhatnak. Nem kell megtanulniuk újabb parancsokat, nem kell foglalkozniuk a környezet megváltozásával. Mivel a Unix a PC-ktől kezdve az ún. nagygépekig nagyon sok hardveren működik, a Unix alatt gépfüggetlen alkalmazói programok is írhatók. Ezt egyrészt a C programozási nyelv, másrészt maga a Unix-környezet teszi lehetővé, amely minden gépen ugyanazokat a könyvtári eljárásokat, függvényeket biztosítja.

A Unix fájlrendszere

A Unixban a fájlok karakterek sorozatát jelentik. Nincs különbség soros és közvetlen elérésű fájl között. Ez a felfogás leegyszerűsíti a fájlokkal kapcsolatos műveleteket, valamint lehetővé teszi azt, hogy a fájlokat és a különböző I/O eszközöket azonos módon kezeljük.

A Unix fájlrendszerében a következő fájltípusok fordulhatnak elő:

— Közönséges fájlok. Ezek a hagyományos értelemben vett adatállományok. Mivel karakterek sorozatából állnak, formájukra nincsen semmi megkötés.

— Könyvtári (directory) fájlok. Ezek megkülönböztetett fájlok, szerkezetük meghatározott. Közönséges fájlok és további könyvtárak katalógusadatait tartalmazzák.

— Speciális fájlok. Ezek fizikai eszközöket jelentenek. Nem igazi fájlok, de kezelésük mindenben megegyezik a közönséges fájlok kezelésével. Például,

ha egy speciális fájlba írunk, egy kezelőprogram „elintézi", hogy az adatok a megfelelő fizikai eszközre kerüljenek.

Maga a fájlrendszer hierarchikus felépítésű: egy feje tetejére fordított fához hasonló. Ebben a szerkezetben fájlok és könyvtárak (directoryk) fordulhatnak elő. A könyvtárakhoz fájlok és újabb (al)könyvtárak tartozhatnak. A fájlok a fa leveleiként foghatók fel, míg a könyvtárak a fa ágainak elágazásait jelentik.

Minden az ún. root (gyökér) könyvtárból indul ki. Ezt a „/" karakter jelöli. A bin könyvtár a végrehajtható fájlok adatait tartalmazza. Az usr könyvtárban többek között a felhasználók könyvtárai találhatók, például a Gabor nevű, amely további alkönyvtárakat tartalmaz. A txt alkönyvtár alatt a cikk1 és cikk2 fájlok találhatók.

A fájlok neve betűket és néhány speciális karaktert tartalmazhat. Hossza maximum 14 karakter lehet. A Unix különbséget tesz kisbetű és nagybetű között, így a prog.prg és a Prog.prg két különböző fájlt jelöl. A néven belül több pont is előfordulhat, azaz a prog.prg.c legális programnév. A fájlokat ez a név csak a saját könyvtárukban azonosítja. A teljes azonosításhoz hozzátartozik az az út is, amely a gyökértől az adott fájlig vezet. Például a Gabor felhasználó txt alkönyvtárához tartozó cikk1 fájl teljes neve /usr/Gabor/txt/cikk1.

Meg kell említenünk, hogy a PC-ken használt MS-DOS fájlrendszerét a Unixéhoz hasonlóan alakították ki. Van néhány — különösen a fájlnevekkel kapcsolatos — eltérés, de az alapvető szerkezet ugyanaz. Aki tehát ismeri az MS-DOS fájlrendszerét, az a Unixéban is könnyen eligazodik.

A shell

A shell egy parancsértelmező program. A Unixnak ugyanolyan fontos része, mint a kernel vagy a fájlrendszer. A shell a Unix felhasználói felülete, segítségével elérhetők és végrehajthatók a Unix segédprogramjai. Programozása külön tudomány, de aki elsajátítja ezt a tudást, az nagyon hatékony eszköz birtokába kerül. (A januári számban elkezdődik a shellprogramozást is ismertető programozói sorozatunk. — A szerk.) A shell lehetővé teszi, hogy a meglévő, általában csak egy adott feladat megoldására készült segédprogramok felhasználásával bonyolultabb feladatokat is meg tudjunk oldani.

A Unix rendszerekben háromféle shell található: a Bourne, a Korn és a C shell. (A Bourne és a Korn az alkotójáról kapta a nevét.) A Unix általános segédprogramjai mind a háromból elérhetők, de további szolgáltatásaik különböznek.

A Berkeley Unixhoz Bill Joy készítette el az ún. C shellt, amit elsősorban a C-ben programozóknak szánt. Olyan többletszolgáltatásokat tartalmaz, amelyek támogatják a hatékony programfejlesztést. Ezek közül a history parancsot említhetjük, amelynek hatására a begépelt parancsok tárolódnak, és újraírás nélkül ismét végrehajthatók.

A Korn shellt David Korn írta az AT&T-nél. Ez a parancsértelmező egyesíti a Bourne és a C shell jó tulajdonságait, de saját ötleteket is tartalmaz. Gyorsabb, mint a C shell, de lassabb, mint a Bourne. Viszonylag új program, de széles körű elterjedése várható.

A felhasználók bejelentkezés után a három shell valamelyikével kerülnek kapcsolatba. (A megfelelőt a rendszeradminisztrátor biztosítja a felhasználónak.) A Bourne és a Korn shell a sor elejére írt dollár- ($) jellel, a C shell pedig a százalék- (%) jellel jelentkezik a terminálon. Ezzel jelzik, hogy készen állnak a parancsok fogadására.

Egy parancs begépelése után a shell a kernel segítségével betölteti és végrehajtatja a parancsnak megfelelő programot. Ha a felhasználó hibásan gépeli be az utasítást, vagy olyan parancsot akar végrehajtani, amelyhez nincsen engedélye, a shell egy hibaüzenetet ír a terminálra. Ezek az üzenetek általában elég jól meghatározzák a hiba jellegét.

A háromféle shell a Unix újabb változataiban is elérhető lesz. Ennek az a praktikus oka, hogy az egyik shellt jól ismerő felhasználó ne kényszerüljön valamely másik használatára.

Déri Gábor

Történeti áttekintés

A Unix operációs rendszert Ken Thompson, Dennis Ritchie és munkatársaik fejlesztették ki az AT&T Bell Laboratories-nál. Első változata egy DEC PDP-7 számítógépre készült 1969-ben. A rendszer használható volt, így 1971-ben az ún. első kiadáshoz (First Edition) már kézikönyv is készült, amelyet Thompson és Ritchie írt. Ebben a könyvben írták le először azokat a lényeges alapelveket, amelyek mind a mai napig jellemzik a Unixot.

1972-ben megszületett az ún. második kiadás (Second Edition). 1973-ban a kernelt és az I/O rendszert a C programozási nyelven újraírták. Ez jelentős változás volt a Unix életében, mert ez tette lehetővé, hogy a rendszert más számítógépeken is használhassák. Az 1975-ben megjelent Sixth Edition már kereskedelmi forgalomban is kapható volt. Ebben az időben a Unix fő felhasználói és támogatói az egyetemek és egyéb oktatási intézmények voltak, amelyek csekély térítés ellenében jutottak az új operációs rendszerhez. Hozzávetőleg 1300 egyetemen használják ma is a Unix valamilyen változatát.

A Unix fejlesztése nem állt meg. 1979-ben megjelent az általános használatra szánt hetedik kiadás. Ebből a változatból származik a BSD (Berkeley Software Distribution) Unix, amelyet Berkeley-ben a University of California fejlesztett ki. Ez a Berkeley Unix néven is ismert változat ugyanolyan népszerűségre tett szert, mint az eredeti AT&T forráskódon alapuló rendszerek. A Unix harmadik ágát jelentették az olyan — 16 bites minigépekre írt — változatok, mint a XENIX, amelyet a Microsoft és az SCO (Santa Cruz Operation) közösen fejlesztett ki.

1980 végén az AT&T bejelentette a Unix System III-at. Ez volt az első lépés a Unix tényleges termékké válásához vezető úton. Miután az AT&T így kilépett a piacra, kibocsátotta a Unix System V Release 2-t. Ez volt az első olyan verzió, amelynek felhasználói a gyártó teljes támogatását élvezték. Az AT&T ezzel a verzióval a Unix szabványosítását is elő akarta segíteni.

1985-ben jelent meg a System V Release 3 verzió, amely a mai napig használatban van. A nagy teljesítményű 386-os, 486-os PC-ken elterjedt rendszerek (SCO, Interactive, stb.) is ezt a verziót valósítják meg. A következő lépés a System V Release 4, amelyet az AT&T mellett már több PC-s Unix is kínál.

INFORMÁCIÓKÉRÉS: 22 ▲

Praktikusan és divatosan

Mint ahogy a Clippert bemutató sorozat előző részeinek témájából ki lehetett találni, ebben a részben az adatmegjelenítést segítő tbrowse és tbcolumn objektumokról lesz szó. Mivel azonban a témához több információ tartozik, mint ami egy részben elférne, kettévágtuk: most áttekintjük az ilyen típusú megjelenítés főbb jellemzőit, egy hónap múlva pedig a konkrét nyelvi lehetőségekkel foglalkozunk.

A Clipper programozási nyelv elsősorban különböző adatstruktúrák kezelésére szolgál. Ezen a témakörön belül az alkotók igyekeztek a lehető legkomplexebb szolgáltatást nyújtani. Ezek közé a szolgáltatások közé tartozik a különböző felhasználói felületek kialakíthatósága is. Az adatbázis-kezelő rendszerekben az adatok megjelenítésének lehetősége jelentősen befolyásolja a program szolgáltatásainak külalakját.

Az adatábrázolásnak alapvetően két módja terjedt el leginkább. Az egyik a grafikus ábrázolás, ami grafikonok, diagramok és más ábrák segítségével igyekszik áttekinthetővé tenni az adathalmazokat. Ezt a lehetőséget a Clipper grafikus könyvtár híján fájdalmasan nélkülözi. Ennek ellenére mégis van egy járható útja, hogy az extend könyvtár használatával megvalósítsunk egy grafikus interfészt, de ezzel a témával majd egy későbbi részben fogunk foglalkozni. Az adatok megjelenítésének másik eszköze a táblázat. Ez nem annyira áttekintő jellegű, mint a grafikus módszer, de lehetőséget ad az adatbázist felépítő rekordok, és az ezeket felépítő mezők egyenkénti elérésére. A Clipper ez utóbbi módszert támogatja.

Sark(alatosság)ok

Az előző verziókból már jól ismert BROWSE( ) és DBEDIT( ) függvények az új rendszernek is részét képezik, de a bevezetett két adatmegjelenítési osztály objektumaival sokkal igényesebbre tervezhetjük programjaink felhasználói felületét. Ezen objektumosztályok használatával felépített táblázatok legfontosabb jellemzői a következők.

A táblázat az általunk megadott négy sarokpont közti képernyőterületen látható. A táblázat sorait az egyes adatrekordok adják. Ezek a rekordok nem feltétlenül egy adatbázis rekordjai, hanem lehetnek bármilyen más legális változók. A táblázat oszlopait a rekordot alkotó mezők adják. Ahány adatot jelenítünk meg soronként, annyi oszlopa van a táblázatnak. A táblázat sorainak és oszlopainak száma kötetlen, de egyszerre csak annyi mező kerülhet ki, amennyi a négy sarokpont közé befér.

Az adatmegjelenítési objektumnak saját belső logikai kurzora van. Ez a logikai kurzor nem egy képernyőpozícióra, hanem a táblázat egy mezőjére mutat. Ezt a kijelölt mezőt aktuális mezőnek nevezhetjük. Az aktuális mezőt az objektum exportált eljárásaival akár abszolút, akár relatív helyre pozicionálhatjuk. Ha olyan mezőre irányítjuk az objektum belső kurzorát, amely éppen nincs kijelezve, akkor az objektum gondoskodik a kijelzés aktualizálásáról. A megjelenítendő mezők tartalmát az általunk megadott kódblokkok visszatérési értéke szolgáltatja. Az aktuális mező kiírásához külön színpárost adhatunk meg, és az alászínezés automatikusságát egy logikai változón keresztül tilthatjuk, illetve engedélyezhetjük.

Színes kódok

Az egyes mezők színezését teljesen az ellenőrzésünk alatt tudjuk tartani: elvben annak sincs akadálya, hogy minden mezőnek más színt adjunk. A színezést ugyancsak kódblokkok határozzák meg. Minden mező kijelzése előtt végrehajtásra kerülnek ezek a kódblokkok, amelyek paraméterként kapják a kijelzendő értéket, és visszatérési értékük határozza meg a mező színeit. Így a negatív számok, vagy más, kiválasztott adatok eltérő színben láttathatók. Lehetőség van minden egyes oszlopnál a fejléc és lábszöveg megadására, továbbá az egyes oszlopokat, valamint a fej- és lábrészt elválasztó karakterek előírására is. Itt kell megjegyezni, hogy mind a fejléc, mind a láb szövegét egy karakteres változóban kell elhelyezni, kifejezés megadására nincs mód. Így ha a lábrészben például a numerikus oszlopösszeget szeretnénk kiíratni, akkor erre más megoldást kell keresnünk.

Rendelkezésünkre áll ezenkívül a BROWSE() függvényből ismert freeze funkció is. Ez a funkció a táblázat első x oszlopába nem engedi beléptetni az objektum logikai kurzorát. Az x értékét természetesen nekünk kell megadni. Ezt a funkciót használhatjuk az adatsorok számozására. Az az oszlop, amelyet a freeze funkcióval befagyasztunk, ugyanúgy kijelzésre és szükség szerint aktualizálásra kerül, mint a többi oszlop, de a logikai kurzor nem pozicionálható rá. Kivétel nélkül minden jellemző megváltoztatható az adatmegjelenítés közben, dinamikusan.

Szabad a kezünk...

Az adatok megjelenítése közben a klaviatúra kezelése teljes mértékben ránk van bízva. Ez lehetőséget ad a különböző navigációs műveletek ellenőrzésén kívüli, egyéb akciók végrehajtására is. Ilyen lehet például egy újabb táblázat megjelenítése is. Az egy időben kijelzett táblázatok száma nincs korlátozva. A táblázat kijelzését fel is függeszthetjük a különböző akciók végrehajtásának ideje alatt.

Kijelzés közben mindig pontosan ismerjük a táblázat kurzorának helyét, és az általa mutatott mező tartalmát. Ennek az értéknek a módosítására kiválóak az előző részben megismert GET objektumok. Erre mutat egyszerű példát a rendszer részét képező \source\sample\tbdemo.prg. A program a paraméterként megadott adatbázist táblázatos formában megjeleníti, és lehetőséget ad az aktuális mező szerkesztésére is. A táblázatot adatmegjelenítési objektumok segítségével rajzolja ki, de az adatok változtatására sajnos a GET és READ makrókat használja az objektumok közvetlen kezelése helyett. Ugyanennek a feladatnak egy sokkal igényesebb és részletesebb megoldását mutatja be a \source\dbu\dbuedit.prg.

Fridl György

Modula-2

Halmozzuk az élvezeteket!

A matematikában a halmazt tetszőleges természetű elemek valamilyen tulajdonsággal jellemzett összességeként definiálják.
Például a prímszámok halmaza, egy egyenes pontjainak halmaza, események halmaza, Magyarország megyéinek halmaza stb.
Halmazokat képezni nem egy ördöngös feladat — az egyetlen kritérium: minden vizsgált objektumra eldönthető legyen, hogy az eleme-e a halmaznak vagy sem.
(Ez talán triviálisnak tűnik, azonban aki már foglalkozott a dologgal, tudja, hogy nem is mindig olyan kézenfekvő.)

A Modula-2 SET típusai lehetővé teszik a programozó számára halmazok létrehozását és kezelését a halmazelméletből ismert műveletek segítségével. A SET-eknek azonban a matematikai halmazokhoz képest komoly korlátai vannak. A halmazok elemeinek alaptípusa itt csak sorszámozott, illetve intervallumtípus lehet, és egy halmaz kizárólag azonos alaptípusú elemeket tartalmazhat. Ezenfelül a Modula-2 halmazok számossága korlátozott. A Wirth-féle definíció konkrétan nem rögzíti ezt a korlátot, csak az adott processzor gépi szavának többszörösét említi mint lehetséges felső határt. Az implementációk nagy része a CARDINAL típus számosságát (PC-n 65536) tekinti a programozó által definiált halmaztípusok számossága felső határának. Sok esetben azonban csak a gépi szó bitszáma ez a határ (ez PC-n 16 lenne).

Természetesen a Modula-2 segítségével kreálhatunk tetszőleges típusú és összetételű halmazokat is, ebben az esetben azonban a halmaz adattípust saját absztrakt adattípusként kell létrehoznunk.

Halmaztípusokat a „SET OF" kulcsszavak segítségével definiálhatunk. A BITSET standard típus definíciója: SET OF [0..W — 1] — ahol W az adott számítógép szóhossza. Ez az előre definiált típus a regiszterek és a memória bitenkénti kezelését teszi lehetővé. Egy adott halmaztípushoz tartozó értéket úgy adhatunk meg, hogy a halmaz elemeit kapcsos ("{", "}") zárójelek között felsoroljuk, és/vagy intervallumokkal definiáljuk, továbbá feltüntetjük a halmaz típusazonosítóját a nyitó kapcsos zárójel előtt. (A kapcsos zárójelek használata megfelel a matematikában elterjedt jelölésmódnak.) Ha az azonosítót elhagyjuk, akkor a fordító automatikusan BITSET típusú halmazt generál.

Az elemek megadása tartalmazhat kiértékelendő kifejezéseket. Az értékek és az intervallumok sorrendje tetszőleges. Mivel a halmazok nem tartalmaz-

Névjegy

Halmaz

A Modula-2 halmaztípusai a matematikai halmazok egyszerűsített változatai. A matematikai definíciónak megfelelően a halmaz nem rendezett — de valamilyen szempont vagy tulajdonság szerint összetartozó — elemek gyűjteménye. A matematikai halmazoktól eltérően a Modula-2 halmazok típusa nem tetszőleges, csak sorszámozott, illetve intervallumtípus lehet. A Modula-2 halmazok számossága ugyancsak korlátozott. A maximális elemszám implementációfüggő, WordLength és MAX(CARDINAL) közötti érték. A halmazokat az implementációk túlnyomó többsége bitfüzérként ábrázolja, ezért a halmazokon végzett műveletek a magasabb szintű absztrakción túl bitmanipulációkra is alkalmazhatók. Ez különösen olyan processzorokon hatékony, amelyek gépi kódú utasítások szintjén képesek a bitek kezelésére. (A legtöbb processzor ilyen — ha nem is bit, de bitcsoport=regiszter/gépi szó/memóriarekesz szinten.) A halmaztípusok hatékony eszközül szolgálnak a programozó számára, mivel lehetővé teszik tetszőleges operandus egyes bitjein végrehajtott műveletek kifejezését a matematikából ismert magas szintű halmazabsztrakció műveleteinek segítségével.

Pontosan és szépen

Definíció:

Egy halmaztípus, melyet SET OF T-ként definiálunk, a T alaptípus értékeinek minden halmazát tartalmazza. T típusa vagy a 0 és N—1 közötti egész számok intervalluma, vagy egy legfeljebb N elemű felsorolt típus, vagy egy felsorolt típus legfeljebb N elemű intervalluma lehet, ahol N az adott implementációtól függő érték.

$ HalmazTípus = SET OF EgyszerűTípus.

A BITSET standard típus definíciója

BITSET = SET OF [0..W -1],

ahol W az implementáció által definiált érték, és az adott számítógép processzorának szóhosszát jelöli.

nak duplikátumokat, ha egy elem többször is szerepel az értékdefinícióban, csak az első előfordulását veszi figyelembe a fordító. Amennyiben a kiértékelésre szoruló kifejezések mind konstansok, akkor a halmazérték konstans definícióként is szerepelhet. Az üres halmazt { }-lel jelöljük.

A Modula-2 halmaztípusai elsősorban a rendszerprogramozók számára jelentenek nagy segítséget a hardverközeli programozásban.

Az az absztrakció, hogy egy elem része-e egy adott halmaznak vagy sem, hardverközeli nyelven azt jelenti, hogy egy „elemedik" sorszámú bit értéke 1 vagy 0. Így például a halmazértékek definiálásakor a programozó minden egyes bit értékét külön-külön tudja meghatározni.

Ez a módszer sokkal előnyösebb, mint mondjuk a kívánt érték megadása decimális vagy akár hexadecimális számként (vö. a C nyelvvel), sőt még a bináris megadási módnál is szemléletesebb és rövidebb (ráadásul bináris számokat a fordítók általában nem is tudnak kezelni). Szemléletesebb, mivel egy hosszú, „0"-ákból és „1"-esekből álló szekvencia helyett az adott értékben logikai „1"-es értéket felvevő pozíciókat kell csak megadni, ami növeli az áttekinthetőséget.

A fentiek miatt persze maga a megadás is rövidebb. Példaként vegyünk egy 32 bites kontrollregisztert, amelynek teszem azt a 9. bitje a printer felé jelzi az adáskészséget. A megfelelő érték decimális, hexa, bináris és BITSET reprezentációja a következő lesz:

```
DEC     256
HEX     100H
BIN
  00000000000000000000000100000000
BITSET     {9}
```

Vagy például próbáljuk megválaszolni ezt a kérdést: a 32 bites regiszter 17. bitje 1-es értékű-e?

```
DEC     786432
HEX     C0000H
BIN
  00000000000011000000000000000000
BITSET     {18,19}
```

Jelölje mondjuk a 20. és a 21. bitek kombinációja a printer spooler rendelkezésére álló puffer méretét az alábbi értelmezés szerint: 00 = 1 bájt, 01 = 512 bájt, 10 = 1024 bájt és 11 = 10 kbájt. A BITSET-es jelölést alkalmazva a másik három jelölésmód kiszámításához és leírásához szükséges idő töredéke alatt tetszőleges pufferkombinációt lehet beállítani. ({9}, {9,20}, {9,21}, {9,20,21}) Az egyedüli szempont, amit figyelembe kell vennünk, hogy a Modulában a bitek sorszáma 0-val kezdődik; ez persze megfelel az általánosan elfogadott konvencióknak, mivel így a bitsorszám a helyi értékre is utal. Az ábrázolásban — szintén a konvenciók és a decimális számrendszerben megszokott mód szerint — a helyi érték és vele a bitsorszám balról jobbra csökken. Természetesen a halmaztípus használatának az előnye nem merül ki az állandó-értékadásoknál; mint látni fogjuk, a Modula halmazműveletei a programozó számára ennél jóval több lehetőséget biztosítanak a bitek manipulálására futásidőben is. Ha ráadaásul nem a BITSET típust használjuk, hanem egy általunk definiált típust, akkor az alkalmazott jelölésmód még egyértelműbben jelzi szándékainkat. Vegyük például a DOS interfész 78-as (4EH) Find First File funkcióját. A híváskor a CX regiszterben meg kell adnunk a keresett fájl(ok) attribútumát. Ez, például egy mezei, DIR-szerű híváskor a 4-es és 5-ös bitek beállításával jár. Ezt C nyelven az alábbi módon csinálhatjuk: CX = 0x30 vagy CX = 48. Modulában ugyanez CX = {4,5}, de ha definiáljuk

Példának okáért

Halmaztípusok deklarációja:

```
TYPE
Colorset = SET OF (Black, Blue, Green, Red, White) ;
NumberSet = SET OF [1..10] ;
Nap = (hetfo, kedd, szerda, csutortok, pentek, szombat, vasarnap) ;
Napok = SET OF Nap ;
InElem = [2..8] ;
IntSet = SET OF IntElem ;
Halmazértékek:
IntSet {2,5}
IntSet {2,4..7}  (* = IntSet {2,4,5,6,7} *)
munkanapok := Napok {hetfo..pentek} ;
NumberSet { }  (* Üres halmaz *)
Colorset {Blue, White}
```

Pascal és C

A Modula-2 halmaztípusai megfelelnek a Pascal halmaztípusainak. Egyedül a halmazértékek jelölésmódja változott úgy, hogy a matematikai jelölésmódhoz jobban igazodjon. (A Pascalban szögletes zárójellel jelöljük a halmazokat.) A C nyelvnek nincsen halmaztípusa. (A bitfield más tészta. Sok esetben hasonlóak az alkalmazási területek, de a koncepció teljesen eltérő.)

Szótár

Halmaz: Tetszőleges objektumoknak valamilyen tulajdonsággal jellemzett összessége. (A halmaz számossága a halmaz elemeinek száma.)
Részhalmaz: Ha egy halmaz minden eleme eleme egy másik halmaznak is, akkor a halmaz a másik halmaz részhalmaza.
Valódi részhalmaz: Ha egy halmaz (A) részhalmaza egy másik halmaznak (B-nek), és B-nek létezik olyan eleme, amely nem eleme A-nak, akkor A valódi részhalmaza B-nek.

Tények, tippek, trükkök C-programozók számára

Nagyon fontos, hogy a bitmanipulációra használandó halmazokban a megadott elemek bitsorszámukkal szerepelnek, nem az általuk esetleg képviselt, mondjuk numerikus értékkel. Így például az „IF 3 IN A THEN ..." kifejezés C nyelven nem „if(3&A)...", hanem „if(8&A).." A különbség abból adódik, hogy míg a Modula-2 bitszinten képes értelmezni a műveleteket, addig a C nyelv bitcsoportra értelmezi azokat. Így természetesen a harmadik sorszámú bit értéke — a helyi értékét is figyelembe véve — numerikusan 8 lesz. Tehát, míg Modulában az elemnek úgymond a sorszámára hivatkozunk, addig a C-ben az elemet nem tudjuk a nagyobb egységtől (gépi szótól) elkülönítve kezelni. Erre azokban az esetekben lényeges ügyelni, amikor a Modulában felsorolt és halmaztípusokat együtt alkalmazunk. Nézzük például a PC billentyűzetének státusszavát (a Modula-2 deklarációk a Keyboard könyvtári modulból valók):

```
TYPE
ShiftFlags  = ( rShift,  lShift,
                Ctrl,  Alt,
                Scroll,  Num,  Caps,  Ins,
                * For enhanced BIOS *)
                lCtrl,  lAlt,  rCtrl,  rAlt,
                ScrollDn, NumDn, CapsDn, SysReq ) ;
KeyboardFlagsSetN = SET OF [rShift..Ins] ; (* XT or normal BIOS *)
KeyboardFlagsSetE = SET OF ShiftFlags ;   (* AT or enh BIOS *)
```

legyen kbStatus[40H:17H] : KeyboardFlagsSetE ;
és mondjuk egy teszt IF {rShift,lShift} <= kbStatus THEN ...
C-ben először a kézenfekvő enum típusdeklaráció jutna az ember eszébe, de ha megvizsgáljuk a halmazokon végezhető műveletek C-beli megfelelőit (a bitcsoport-műveleteket), láthatjuk, hogy célszerűbb numerikus konstansokat deklarálni (a teljesség igénye nélkül):

```
#define rShift 1
#define lShift 2
/*... és így tovább 13 soron keresztül */
#define SysReq 32768.
```

A változók deklarálása: a C-ben nem tudunk fix című változót deklarálni.
És a teszt :

```
if ( (rShift + lShift) & *( (int far)0x417 ) ) ...
```

Ennyi.

a directory=4 és archive=5 konstansokat, akkor a CX={directory,archive} utasítás már mindent elmond szándékainkról. Akár egy új halmaztípust is érdemes létrehozni az ügy érdekében, például

FileAttr = SET OF (readonly, hidden, system, volume, directory, archive)

definícióval. (Természetesen a Modulában is használható a CX=30H, illetve CX=48 jelölésmód.) A további példák az LST és MOD fájlokban találhatók.

Halmazoperátorok

A +, –, *, / műveletek azonos típusú halmazokra alkalmazhatók, és jelentésük a következő (A és B azonos típusú halmazokat jelölnek):

A + B — halmazunió — halmaz, amely minden olyan elemet tartalmaz, amelyek elemei A-nak, B-nek vagy mindkettőnek

A — B — halmazkülönbség — halmaz, amelynek elemei A-nak azon elemei, melyek nem elemei B-nek

A * B — metszetképzés — halmaz, melynek elemei azok az elemek, amelyek mind A-nak, mind B-nek elemei

A / B — szimmetrikus — halmaz, melynek elemei elemei A-nak vagy B-nek, de nem mindkettőnek

Mivel a halmazokat a Modula-2 bithalmazként ábrázolja, a fenti műveletek sorrendben a bitenkénti OR, AND NOT, AND és XOR operátoroknak felelnek meg.

A halmaztípusú változókhoz a típuskonvenciók betartásával (vagy ellenőrzött megkerülésével) értékeket rendelhetünk, akárcsak a már korábban megismert típusok esetén. Azonos típusú halmazok között az alábbi relációk vannak értelmezve:

A = B — TRUE (igaz) ha: A és B pontosan ugyanazokat az elemeket tartalmazzák

A # B — A = B FALSE (hamis)

A <= B — A (nem feltétlenül valódi) részhalmaza B-nek

A >= B — B <= A

Egy speciális relációs operátort az „IN" kulcsszó jelöl. Az IN operátorral tesztelhetjük, hogy egy elem része-e egy adott halmaznak. Ha k egy kifejezést jelöl, amelynek értéke kompatibilis a h halmaz alaptípusával, akkor a k IN h reláció TRUE (igaz), ha a k által reprezentált érték a h halmaznak eleme. Fontos megjegyzni, hogy az IN az egyetlen olyan relációs operátor, amelynek operandusai eltérő típusúak.

A Modula-2 két standard eljárást biztosít a halmaztípusú változók kezelésére. Az INCL eljárás egy halmazba helyez egy elemet, ha az még nem része a halmaznak. A keretes rész definícióit felhasználva, ha A egy Colorset típusú változó, akkor az INCL (A, Blue) utasítás A-ba helyezi Blue-t. Ez ekvivalens az A:= A+Colorset{Blue} kifejezéssel. Hasonló módon az EXCL eljárás egy elemet eltávolít a halmazból, ha az megtalálható a halmazban, ellenkező esetben nem történik semmi. Így az EXCL (A,Blue) kifejezés megfelel az A := A — Colorset{Blue} kifejezésnek.

Mindkét eljárásban az első paraméter egy halmaztípusú változó, míg a második egy tetszőleges kifejezés, amely az első paraméter alaptípusával kompatibilis eredményt produkál. A Modula-2 implementációk egy része az INCL és EXCL eljárásokra hatékonyabb kódot generál, mint az ekvivalens értékadó kifejezések. Ez a halmazműveletekre általában is igaz, mivel egyes processzortípusok képesek bitenkénti manipulációra. (Az Intel 80x86 család nem ilyen.) Mivel a Modula-2 implementációk túlnyomó többsége a halmazokat bithalmazokként (bitfüzérekként) ábrázolja, ezért sok alkalmazásban a halmazokat logikai (Boolean) vektorként használhatjuk. (A vektor hosszát a halmaz számossága határozza meg.) Ezért a bitmanipulációra képes processzorokon a halmazok használata előnyösebb, mint a BOOLEAN tömböké, mivel a halmazok segítségével a futásidő jelentősen csökkenthető (a műveletek az összes vagy több elemen egyszerre végezhetők el). (Példák az LST fájlban.)

Villányi László

KALEIDOSZKÓP

Irreálisnak bizonyultak reményeink, hogy egyszeriben sikerül előrehoznunk lapunk megjelenését a hónap elejére. Pedig emiatt tűztük ki olyan rövidre a határidőt a Kaleidoszkóp első jelentkezésekor. Ezúton is kérjük olvasóink szíves elnézését. Jónéhány „hívünk" ennek ellenére, a szűkre szabott határidőn belül beküldte megfejtését, ezekre következő számunkban visszatérünk.

Már most megállapítható, hogy az ötlet megnyerte olvasóink tetszését: több levél igazolja, hogy sikerült felkeltenünk a kíváncsiságot egyrészt a rejtvényszerű feladatok, másrészt a természetes nyelvek eleddig ismeretlen sajátosságai iránt. Tartjuk magunkat ígéretünkhöz — igyekszünk változatos betekintést nyújtani a természetes nyelvek tarka világába. Ne gondolja hát senki, hogy előzetes nyelvismeret kívántatik a feladatok megoldásához. Éppen ellenkezőleg, az a célunk, hogy ismeretlen, egyébként szinte elérhetetlen nyelvek színes világába invitáljuk az érdeklődőket...

Ha kissé elkésve is, most már elárulhatjuk, hogy első példánkat a thai nyelvből vettük, amely a kínai-tibeti nyelvcsaládnak a kínai után a legnépesebb tagja.

Bizonyára sokakat meglepett, hogy a nemek szerinti megkülönböztetés helyett itt a főneveknek másfajta „típusokba sorolása" található meg. Bármennyire hihetetlen is eddigi idegennyelv-ismereteink alapján: a nemek szerinti megkülönböztetés egyáltalában nem általános a világ egyéb, nem indoeurópai nyelvcsaládjaiban.

Mostani első feladványunk inkább csak ujjgyakorlatnak tekinthető ennek az érdekes megfigyelésnek a finomítására — egy újabb „ismeretlen" nyelv alapján.

Íme, az első feladvány:

Adva vannak számneves kifejezések a jelentésükkel együtt, továbbá egy rövid kiegészítő szójegyzék valamilyen, számunkra egyelőre ismeretlen nyelvből.

A) Kifejezések

dzsúnihiki-no hicudzsi	= 12 birka
dzsútó-no usi	= 10 tehén
gohon-no kasza	= 5 napernyő
gonin-no muszume	= 5 lány
hacsitó-no kuma	= 8 medve
icsitó-no dzó	= 1 elefánt
kuhiki-no uszagi	= 9 nyúl
kuhon-no szao	= 9 pózna
nihiki-no inu	= 2 kutya
nihon-no jubi	= 2 ujj
nimaj-no szara	= 2 tányér
rokumaj-no simbun	= 6 újság
rokuva-no cubame	= 6 fecske
sicsihon-no empicu	= 7 ceruza
simaj-no tidzu	= 4 térkép
szanken-no ie	= 3 ház
szannin-no gisi	= 3 mérnök

B) Kiegészítő szójegyzék

bin	= flaska
csocso	= lepke
empicu	= ceruza
kami	= papírlap
kippu	= jegy
koja	= kunyhó
kuda	= tülök
neko	= macska
nezumi	= egér
rakuda	= teve
szenszej	= tanár
tora	= tigris

Kérdések:

1. Hogy mondják ezen a nyelven azt, hogy 1 ceruza, 2 kunyhó, 3 teve, 4 macska, 5 tülök, 6 tigris, 7 egér, 8 lepke, 9 flaska, 10 jegy, 11 tanár?

2. Mi az eltérés a mostani nyelv és az októberi Kaleidoszkópban megismert nyelv „típusfelfogása" között?

3. Milyen nyelvről lehet szó?

Második feladványunk ugyanebből a nyelvből mutat be egy másféle jelenséget. Megoldása lényegesen nehezebb, hiszen ehhez már komoly szintaktikai meggondolások szükségesek. Az eredmény viszont nem kevésbé meglepő; ezt elszántabb, gondolkodásra kész olvasóink látni fogják.

Íme, a „korpusz":

Ismert mondatok, ismert jelentésükkel:

1. Vatasi-va hon-o jonda. = Könyvet olvastam.
2. Anohito-va zassi-o jonda. = Folyóiratot olvasott.
3. Vatasi-va tegami-o kajta tomodacsi-o miru. = Látom azt a srácot, aki a levelet írta.
4. Vatasi-va tomodacsi-no kajta tegami-o uketa. = Megkaptam a levelet, amit a srác írt.
5. Vatasi-va tegami-o kaku hito-o mita. = Láttam azt az embert, aki levelet ír.
6. Anohito-va vatasi-no jomu hon-o miru. = Látja a könyvet, amit olvasok.
7. Vatasi-no mita hito-va jomu. = Az ember, akit láttam, olvas.
8. Vatasi-va kaku. = Írok.

Feladatok

1. Fordítsuk le a fenti korpusz alapján az alábbi mondatokat. (Vagy nem mind mondat?)

a) Hito-va kajta.

b) Kajta hito-va.

c) Vatasi-va anohito-no kaku tegami-o jomu.

d) Zassi-o jonda tomodacsi-no uketa tegami-o kajta hito-va vatasi-o miru.

2. Mi lenne a d) mondat helyes zárójelezése? Mi teszi lehetővé a megértését zárójelezés nélkül?

3. Hogyan lehetne szintaktikai generáló szabályokkal leírni (és előállítani) a fenti mondatok és kifejezések szintaktikai „csontvázát"?

Végül extra feladat, külön jutalomért:

4. Hogyan lehetne olyan programot készíteni, amely lépésről lépésre szemléletesen bemutatná a generálás menetét? (Tanács: a szemléletesség kedvéért lehetőleg a generálásban részt vevő elemek magyar megfelelőit mozgassuk együtt az idegen nyelvű elemekkel!)

A megfejtéseket a szerkesztőség címére (1251 Budapest, Pf. 71) kérjük beküldeni 1991. január 10-ig. Mivel pedig a múltkori szoros határidő kiírásában sárosnak érezzük magunkat, pótlólag elfogadjuk (és pontozzuk) az októberi számunkban megjelent feladvány megoldását is ugyanezzel a határidővel!

Az értékelésben létraversenyszerűen összesítjük a szerzett pontokat. A legjobb megoldásokat közölni szeretnénk az Alaplapban, illetve mágneslemez mellékletünkön. Az „éllovasokat" félévenként az Alaplap, illetve a Floppy.Lap éves előfizetésével díjazzuk. Az extra feladatokért külön díjazás jár: 3-3 szabadon választott sharewarelemez!

A létraverseny állása: Csurgay Péter 30, Dezső András 30, Geszat Zoltán 30, Süle Gábor 40, Vágó Dániel — Láng Attila D. 30.

Vargha Dénes

„Egyéb javaslatai, észrevételei..."

Az Alaplap alapkérdéseiről

Az Alaplap „vízrebocsátásakor", 1990 júniusában azt mondtuk: másfél év múlva lehet először reálisan mérlegelni, hogy mennyire vált be új lapkoncepciónk: továbbhajózhatunk-e vele, vagy meg kell azt változtatni, ha a felszínen akarunk maradni. Hát letelt a másfél év, és közvéleménykutatásunkra támaszkodva most igyekszünk szembesülni az olvasók igényeivel. Ez alkalommal a lap kiadásának és szerkesztésének alapkérdéseit boncolgatjuk.

Kezdőknek vagy haladóknak

Eredeti elképzelésünk szerint az Alaplapot a számítógépek használatához bizonyos (közepes) hozzáértéssel felfegyverzett, de nem számítástechnikus alkalmazók széles táborának szántuk. Az első meglepetés akkor ért bennünket, amikor azt tapasztaltuk, hogy a profi számítástechnikusok köréből egyre többen válnak rendszeres olvasóinkká. Elkezdtünk tehát közölni „profibb" programozástechnikai témákkal foglalkozó cikkeket is, és azok kedvező fogadtatásra találtak. Ezek után jött a második meglepetés: a közvéleménykutatási kérdőív spontán véleménynyilvánítási rovatában legaktívabban a „kezdőcsoport" deklarálta igényeit. Csak néhány jellemző idézet:

— „Az Alaplap gondoljon a számítástechnika alkalmazói között lévő kezdőkre... és terjessze ki ismeretnyújtó tevékenységét az alapfogalmakra."

— „Úgy érzem, lapjukat a haladóknak és a profiknak szánták. Gondoljanak azokra is, akik a közelmúltban kerültek a számítógép közelébe!"

— „A kezdőknek nagyon kell kapaszkodni. (De megéri.)"

— „Komoly, magas szakmai színvonalú lapra van szükség, de olykor jó lenne egyes problémákról a számítástechnikával most ismerkedők számára is közérthetően írni!"

S hogy hol kezdődik a kezdő? Íme ehhez egy szombathelyi olvasónktól származó adalék: „Mivel inkább szoftverhasználó vagyok, szívesen venném, ha közérthető leírások lennének a különböző szoftverekről, mint például a Word szövegszerkesztő programról volt, amelyet még a 7 éves fiam is 2 nap alatt elsajátított." (Vagyunk olyan önkritikusak, hogy ez inkább az ő fiának képességeit dicséri, mintsem a mi ismertetőnkét.)

Az érem másik oldala viszont, hogy a második legerősebb „lobby" éppen a profi számítástechnikusoké, akik konkrét témajavaslatok tömegével árasztottak el bennünket, az Assembly tanfolyamtól a tárgyorientált programozásig.

Következtetésünk: Fenn kell tartani az Alaplap „többszintűségét", békesen egymás mellett kell megjelennie az alapfogalmak ismertetésének (és gyakoribb ismétlésének) olyan anyagokkal, amelyekből egyik olvasónk egyelőre még „csak a névelőket és a kötőszavakat" érti. Azt is láthattuk a kérdőívre adott válaszokból, hogy a legjobb profi is lehet valamilyen másik témában teljesen kezdő és szívesen fogadja, ha abban őt oktatják. (Mellesleg, és aki nem tudná: ettől ő a profi!)

Hardver vagy szoftver

Ebben a kérdésben is okoztak nekünk az olvasók némi meglepetést. Az Alaplapot szerettük volna kifejezetten szoftver-alkalmazói folyóirattá tenni, mondván, hogy a hardverről a többi lap bőséges tájékoztatást ad. Sikerült elég egyértelművé tenni szoftverorientáltságunkat, s azt hittük, hogy az olvasók ezt természetesnek veszik. Hát nem egészen! Akadt persze, aki azt írta, hogy „Ez egy alapvetően szoftverújság, maradjon is meg annak, így jól kiegészíti a többit."

Azok közül azonban, akik ebben az ügyben véleményt nyilvánítottak, túlnyomó többségben voltak a hardveres témák hiányolói. Ezen belül az igények elég széles skálán mozognak, a hardver-alapismeretek oktatásától kezdve a hardverteszteken át egészen a kapcsolási rajzok közléséig. Szerencsénkre senki nem úgy fogalmazott, hogy mondjunk le a lap szoftveres jellegéről, csupán sokan szeretnék, ha palettánkra felvennénk hardverrel foglalkozó anyagokat is. Eredeti koncepciónkon tehát csak kicsit kell korrigálni, s azt az olvasói igények pontosabb ismerete alapján már el is kezdtük.

Gépfüggőség

Teljesen egyetértünk egyik olvasónkkal, aki azt írja, hogy „a Mikroszámítógép Magazin megszűnése nagy űrt hagyott maga után!" Ennek bekövetkezését akkor is előre láttuk, amikor 1990 tavaszán a lap fennmaradása érdekében a profilváltás kényszerű lépéséről kellett döntenünk. Sovány vigasz a lap nélkül hagyott rétegnek, hogy mi őszintén együttéreztünk velük, de utólag is beigazolódott: nem volt más kivezető út, s az Alaplap akkor kidolgozott koncepciója az eltelt másfél év gyakorlati próbáját kiállta. Kedves olvasóinknak, akik a C-64, Sinclair, Enterprise stb. gépek programjaival kapcsolatos írásokat kérik, sajnos semmit nem tudunk ígérni. De még a PC-khez hasonló vagy nagyobb tudású, csak éppen más felépítésű és más operációs rendszerű mikrogépek használóinak sem (Amiga, Apple). Kezdettől fogva láttuk és eleve elkerültük azt a csapdát, amit egyik olvasónk így fogalmazott meg: „Sokáig nem tartható fenn, hogy a lap mindennel foglalkozzék (Unix, DOS, Commodore, Apple stb.). Legyen specifikus!"

A gépfüggőség dilemmáját sajnos még ezzel az „öncsonkítással" sem oldottuk meg. Ugyanis az IBM-kompatibilis gépek családjában sincs teljes harmónia. Ott vannak például a kisebb tudású, winchester nélküli XT-k, bizonyos programok használatában pedig az AT kategórián belül is állandóan

korlátokba ütközünk, főleg a videokártyák és képernyők nem éppen kompatibilis „hierarchiája" miatt (Hercules, CGA, EGA, VGA stb.).

Lemezmelléklet

Meglepően sokan és meglepően „sokféleképpen" nyilvánították véleményt a lemezmelléklettel kapcsolatban. A többség fenntartás nélkül dicsérte az ötletet és annak megvalósítását, egy nem jelentéktelen kisebbség azonban éles kritikával illette. A panaszok részben technikai jellegűek, részben tartalmiak voltak. A tanulságok levonása és a hibák kiküszöbölése érdekében mi most nem az elégedett hangokból idézünk, bár azok voltak többen.

— „A mellékletet winchester nélküli, monokróm gépen nem tudom elolvasni."

— „Szeretném, és szerintem jobb lenne, ha a mágneslemez melléklet rendszerlemez lenne."

— „A lemezmelléklet színvonaltalan, így felesleges, csak a költséget növeli. Mint reklámhordozó is kicsi lehet a hatékonysága."

Az itt helyhiány miatt nem idézett számos kritikából az is egyértelműen kiderült, hogy az olvasók egy része kezdi a mágneslemez mellékletet önálló műfajként kezelni és annak szerkesztését a hagyományos sajtóhoz hasonló szigorral értékelni. Ehhez bizony mi még nem nőttünk fel (sőt a kezdeti időszakhoz képest visszaestünk!), de az észrevételek alapján 1992-ben néhány „fazonigazítást" mindenképpen elvégzünk a lemezen. Azokat, akik az állandó helyhiány miatt az 1,2 megabájtos HD lemezt javasolják, szeretnénk emlékeztetni a költségnövekedésnél is nagyobb akadályra, hogy a csupán 360 kilobájtos floppymeghajtóval rendelkezők aránya ma még igen nagy.

Külcsín — belbecs — ár

Ezekben az egymással szorosan összefüggő kérdésekben az Alaplap olvasótábora úgy oszlik meg, hogy a többség számára elfogadható a jelenlegi nyomdatechnikai megoldás. A két ellenpóluson elhelyezkedő kisebbség is hallatta persze a hangját. „A gazdag tartalomhoz képest kissé szegényes a külső" — mondják az egyik oldalon, s nekik az egész lap gazdagabban illusztrálva, színesen, fényes papíron kellene, még ha többe is kerülne. Az ellentábor szerint viszont nem kell a lapot „csicsászni", legyen az inkább egyszerűbb és olcsóbb. Még a jelenlegi felemás megoldással kiegyező többségen belül is sokan képviselték azt az álláspontot, hogy az Alaplap „egyetlen rossz tulajdonsága az ára, de reméljük, megoldható, hogy ne emelkedjék tovább."

Valóban drága lenne az Alaplap? Nézzük meg például a most lezárult 1991-es évfolyamot. Ha az olvasó a mellékletként kapott 12 Polaroid lemezt önmagának „anyagáron", 60 forintjával elszámolja, s hozzáadja a novemberi ráadás, a játékmelléklet értékét, akkor a lap megvásárlására fordított összegből „áruban" máris megtérült mintegy 1200 forintja. A fennmaradó összeg egy példányra eső költsége a lap postai beszerzése (előfizetése vagy példányonkénti vásárlása) esetén 96 forint, a PC Turbo Klub tagjainak pedig 76 forint. Tisztán ennyiért jut hozzá az Alaplap teljes információtartalmához, közvetlenül hasznosítható segédprogramjaihoz, hirdetési információkérési lehetőségéhez stb. Kap-e az olvasó ezért a pénzért megfelelő ellenértéket?

A válaszokból az derült ki, hogy az Alaplap az olvasóközönség zömének pénztárcájához mérten „kicsit drága, de megéri." Mellette természetesen mindkét irányban szerepeltek kisebbségi álláspontok: felfelé az, hogy minőségjavítással (több színes nyomás, jobb papír, nagyobb terjedelem, alternatív floppyfajta stb.) még 3-500 forintos ár is elfogadható lenne, lefelé pedig az, hogy inkább rontsunk a kivitelen, ne tegyünk bele lemezt stb., de legyen olcsóbb a lap. Mi a többség véleményét reálisnak tartjuk, s amíg nem kényszerülünk rá, árat sem emelünk.

Összkép

Az Alaplap koncepciójának lényeges eleme, hogy a profilváltás és az olvasótábor egy részének szükségszerű kicserélődése ellenére megőrizze és „újratermelje" a Mikroszámítógép Magazin és annak rendszeres olvasói között évek során kialakult közvetlen viszonyt. Aki ismeri a véleménykérő kérdőívek szakmai titkait, az tudja, hogy 5-10 százalék közötti visszaérkezési arány már jó eredménynek számít. A mi esetünkben a közel 2100 kitöltött kérdőív a ténylegesen értékesített példányszámhoz képest meghaladja a 20 százalékot. Az egyéb vélemények, javaslatok rovatát pedig 900-an töltötték ki, túlnyomórészt valóban hasznosítható ötletekkel, javaslatokkal fejezve ki, hogy sajátjuknak érzik az Alaplapot. Nehezen képzelhető el, hogy ekkora aktivitás csak a Mikropo által főnyereményként felajánlott AT-nak és a Polaroid jutalmaknak köszönhető — de az sem baj, ha ilyen ösztönzőkkel sikerült néhány százaléknyival teljesebbé tenni a felmérést, a nyertesek listáját pedig januári számunkban közöljük.

Távol áll tőlünk, hogy túlbecsüljük az Alaplappal kapcsolatban megnyilvánult — s néha egészen megható módon kifejezett — lelkesedést, ragaszkodást, felértékelést. Tudjuk, hogy ilyen esetekben szükségképpen dominálnak a pozitív vélemények, mert akik számára a lap nem megfelelő (vagy nem ismerik), azok eleve nem is olvassák. Az ő távolmaradásuk is vélemény, ami a felmérésben nehezen dokumentálható! De másfél éves „erőltetett menet" után — valljuk be! — jól esett olvasni munkánk eredményéről olyan kis tömör véleményeket, hogy „Ez egy jó újság." Sok válaszadó pedig igencsak kritikus megjegyzései közé építette be — és tette számunkra ezáltal még inkább hitelessé — pozitív véleményét.

Végezetül ezekből is idézünk néhányat:

— „Induláskor sajnáltam a Mikromagazint, de mára az Alaplap felnőtt saját magához."

— „Valamennyi eddigi példányt visszamenőlegesen olvastam el, amikor egy ismerősömnél rábukkantam. Nem lehetett napokig (hetekig?) hozzám szólni!"

— „Ha munkahelyem lemondaná a lap előfizetését, akkor is megvenném."

— „Az Alaplap áll legközelebb a hétköznapi magyar számítástechnikai valósághoz."

— „№ 1 számítástechnikával foglalkozó újság az Alaplap."

— „Ha a lap érkezik, a barátnőm sem érdekel aznap."

Leszámítva az itt most nem is idézett túlzó dicséreteket, számunkra ez a legutóbbi „vallomás" volt a legkedvesebb. (És olvasónk barátnőjének legnagyobb szerencséjére távol állunk attól, hogy hetilappá vagy napilappá váljunk!)

A kitöltött kérdőíveken található javaslatok, észrevételek gazdag tárházába még jó ideig rendszeresen visszajárunk ötletekért, de most, a lap egészével kapcsolatos vélemények összesítésekor érezzük természetesnek, hogy valamennyiüknek őszintén köszönetet mondjunk a lelkes együttműködésért. Válaszaikban nem éreztük formálisnak a hozzánk intézett sok jó szót és gratulációt, ezért mostani „kollektív viszontválaszunkban" bizonyára Önök is megérzik, hogy köszönetünk mennyire személyüknek szól és szívből fakad.

Faklen Pál

Moduláris számítástechnikai tananyag — naprakészen

„A hónap témája" felvezetéseként felsorolt gondok megoldásának égető szükségességét szem előtt tartva — számos alkalommal, különböző fórumokon számtalan gyakorló szaktanárral konzultálva — az ÉGSZI/Scola Alapítvány (teljes nevén: „ÉGSZI/Scola" Oktatási Intézményeket és Szervezeteket Támogató Alapítvány) egy programot kezdeményezett. Ennek célja moduláris, rugalmas, mindig naprakész és az akár iskolánként eltérő feltételekhez igazodó számítástechnikai tananyagrendszer kialakítása és bevezetése. A tananyag egyes részletei, moduljai füzet formában jelennek meg, ezért a sorozat neve: Számítástechnikai Oktatási Füzetek.

A füzeteket — eredményes tapasztalataik felhasználásával — aktív szaktanárok írják, és a lektorokat is ebből a körből kérik fel a bírálatra. Ez év március végén látott napvilágot az első, amelyet szeptemberig újabb tizennégy követett; további tizenöt kiadásának előkészületei folynak.

A nyári hónapokat leszámítva négy hónap alatt, mintegy négyszáz (középfokú és általános) iskolában közel 30 000 db került a tanárok és tanulók kezébe, egyértelmű elégedettséget jelző visszhanggal. E sorozatból válogathatnak az ország számos helyén indított számítástechnikai szaktanfolyamok, és mind többen keresik az anyagokat magántanulási célból is.

A tanévkezdésig az itt sorolt kiadványok jelentek meg:

1. DOS alapismeretek (N. Bodor Erzsébet — Pálmai Éva)
2. és 3. Pascal feladatgyűjtemény I., II. (Szíjártó József)
4. Turbo Pascal grafika (Huszár Gábor)
5. A Norton Commander segédprogram használata (D. Horváth Erzsébet)
6. Norton segédprogramokról kezdőknek (Melkovics Gyula)
7. PC Tools 5.5 verzió (Keczeli László)
8. Mágneslemezes adatállományok (Gyetvai Károly)
9. Assembly és gépi szintű programozás Commodore számítógépen (Gyetvai Károly)
10. Táblázatkezelők — Lotus 1-2-3 (Nemcsik János)
11. Az adatbázis-kezelés alapfogalmai (Horváth Katalin)
12. és 13. A dBase III. Plus használata I. és II. (Horváth Katalin)
17. Hálózati ismeretek (Horváth Katalin)
18. Ékszer 3.52 grafikus szövegszerkesztő és adatbázis-kezelő rendszer (Huszár Gábor)

A füzetekből minden iskola (tanfolyam) saját tantervéhez, tematikájához igazodóan állíthatja össze az adott helyen kívánatos tananyagot. Ezzel kiküszöbölhetik az oktatók a tanóra-kihasználás csökkenésével járó hosszadalmas diktálásokat, a tanulóknak mindig kéznél van a — jegyzetelési képességeiktől független — aktuális tananyag, s nem utolsósorban: nem kényszerülnek a tanuláshoz csak egyes részleteikben alkalmas, igen drága szakkönyvek megvásárlására.

Fontos körülmény, hogy a tankönyvkiadás hagyományos gyakorlatától eltérően e füzetek — kellő számú (500 pld.) igény beérkezésétől számítva — igen rövid átfutási idő (6-8 hét) alatt már az érdekeltek kezében vannak. Ez a gyorsaság annak is köszönhető, hogy a sorozatban megjelenő füzetek anyagai a kiadásra felajánlásuk időpontjában lényegében kidolgozottan várnak a szerző szaktanárok fiókjában. Az Alapítvány ezt feltételként ki is köti; így biztosítva a gyorsaságon kívül azt is, hogy a gyakorlatban tartalmilag és didaktikailag egyaránt kiérlelt tananyagrészek jelenjenek meg.

Az ÉGSZI/Scola Alapítvány a füzetek megjelentetését teljesen nyitott sorozatként szervezi: ahhoz bármikor, bármely témakörben elfogad — sőt: kér! — szerzői bejelentkezéseket (a már említett módon „kész" anyaggal rendelkező szaktanároktól).

Az ismertetett kísérlet alig egy fél év elteltével bizonyította hasznosságát, életrevalóságát, és az iránta megnyilatkozó, fokozódó érdeklődés gazdag folytatásra enged következtetni.

Előkészületi stádiumban vannak a következő témák:

Programozás dBase-ben I/II/III; Novell alapismeretek kezdő felhasználók részére; Számítástechnikai alapismeretek; A számítástechnika története; EMCO F1 CNC marógép kezelése és programozása; Példatár a Compact 5 CNC eszterga programozásának oktatásához; Példatár Hunor 731 vezérléssel készíthető programokhoz; A Minitum CNC esztergagép programozása; Oxfort Pascal példatár és programgyűjtemény; Quick Basic 4.0 verzió; Üzleti grafika és Harvard Graphics kezdők részére; Relációs adatbázisok; Szakértőrendszerek.

Berencsi István

Bibliográfia

Ez alkalommal is a hónap vezértémájához kapcsolódó könyvek között válogattunk. Olvasóink figyelmébe ajánljuk a lapunk 1990. szeptemberi számában közölt hasonló témájú összeállításunkat is.

Bárdos Attila — Körtvélyesi Gézáné: Programozási alapfeladatok gyűjteménye.
Budapest, 1985. SzÁmalk, 209 oldal. Ára: 101,– Ft.

Dienes professzor játékai.
Budapest, 1989. Műszaki Könyvkiadó, 160 oldal + melléklet. Ára: 98,– Ft.

Fizika és számítástechnika III. Elektromágnesesség, Optika, Atomfizika, Csillagászat. (Sulikomp sorozat)
Budapest, 1990, Novotrade Kiadó, 196 oldal. Ára: 149,– Ft.

Kocsis Zoltán — Gazsó Zoltán: PC-START. (Az IBM-kompatibilis személyi számítógép kezelése kezdőknek.)
Budapest, 1991. Innotech Kft, 99 oldal. Ára: 195,– Ft.

Lőcs Gyula: A Basic és a kíváncsi.
Budapest, 1985. Tankönyvkiadó, 235 oldal. Ára: 14,– Ft.

Matematikai-számítástechnikai szigorlat. (Írásbeli feladatok gyűjteménye.)
Budapest, 1989. SzÁmalk, 163 oldal. Ára: 150,– Ft.

Mérő László: Észjárások.
Budapest, 1989. Akadémiai Kiadó — Optimum Kiadó, 274 oldal. Ára: 148,– Ft.

Racskó Péter: Bevezetés a számítástechnikába.
Budapest, 1989. SzÁmalk, 250 oldal. Ára: 380,– Ft.

Szűcs Pál: A számítógépes oktatás hatékonyságvizsgálata.
Budapest, 1991. OMIKK, 233 oldal. Ára: 196,– Ft

Németh János

Turbo C 2.0 fejlesztőrendszer

(Budapest, 1991. LSI Oktatóközpont, 224 oldal. Ára: 339,— Ft.)

Mialatt a Borland cég méltán népszerű Turbo Pascal rendszeréről elég sok könyv jelent meg az utóbbi években, a szintén igen kitűnő Turbo C egy kissé el lett hanyagolva. Éppen ezért nagy várakozás előzte meg ezt a könyvet, hiszen az olvasót ennek a rendszernek a rejtelmeibe hivatott bevezetni.

Németh János könyve azoknak nyújt nagy segítséget, akik már ismerik a C nyelvet, és szeretnének profi Turbo C-programozókká válni. A könyv nem a C nyelvet oktatja — ilyen témával már elég sok kiadvány foglalkozik nálunk is —, hanem a Turbo C-t mint környezetet. A könyv szerkezete világos, áttekinthető, bár kissé eltér a megszokottól. Ez inkább előnyére válik, ugyanis mielőtt bevezetne a gyakorlati alkalmazásba, megadja azokat az IBM PC-specifikus ismereteket, amelyek nélkül a gépen C nyelven programozni egy kissé 'vakrepülés'. A könyvben ki vannak emelve azok a részek, ahol Kernighan-Ritchie által definiált C és a Turbo C között eltérés tapasztalható.

A szerző példaprogramokon keresztül igyekszik megismertetni a TC 2.0 lehetőségeivel, témakörönként különválasztva a fájlkezelést, a grafikát stb. Ezek a programok áttekinthetőek, igaz, nem tartalmazzák mindig a legprofibb „trükköket", de ez a kezdőknek inkább előny, mint hátrány: ugyanis ezáltal a program számukra könnyebben olvashatóbbá válik. Nem maradhatott ki természetesen magának a program kezelésének az ismertetése sem. A Turbo integrált fejlesztői környezet leírásán túl a parancssorból történő fordítás és a Turbo segédprogramok közül a TCINST, TLINK, TLIB, BGIOBJ is helyet kaptak, sőt a könyv végén megtalálható a TC 2.0 függvényeinek leírása is.

Mindezek mellett a kiadványnak vannak „árnyoldalai". Ilyenek például azok a nyomdahibás részek, ahol a könyv egyes helyeken olvashatatlan. Sajnos egy szakmai hibát is találtam a példaprogramok között: a 67. oldalon az adatbázis-struktúrát kiíró programban a fájlt binárisnak kell megnyitni (fopen(argv[1],"r+b")). Mindezek mellett a könyv remélhetőleg sokaknak nyújt segítséget, és tovább bővíti a magyar Turbo C-programozók táborát.

Goda Attila

NÉMETH JÁNOS
TURBO C
2.0
FEJLESZTŐRENDSZER
LSI OKTATÓKÖZPONT

A Mikrobazár rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk.
Az 1992. februári számtól érvényes díjszabás: kereskedelmi tevékenységet szolgáló hirdetésekért gépelt soronként (60 karakter) 300 Ft. Kérjük, hogy a hirdetés díját a Cédrus Rt.-nek a Budapest Banknál vezetett 380-66760 számú számlájára utalják át, vagy postautalványon a Cédrus Rt. címére (1251 Budapest XI., Karolina út 17.) fizessék be, a hátoldalán feltüntetve, hogy apróhirdetés. A befizetést igazoló szelvényt a közlendő hirdetési szöveggel együtt az Alaplap szerkesztőségéhez küldjék el: 1251 Budapest, Pf. 71. Nem kereskedelmi célú, egyéni hirdetések közlése INGYENES!

ADOK

AMIGA 500-ba belső, külső memóriabővítés, CHIP-FAST RAM átkapcsolóval. AMIGA 500-as vadonatúj 0 km-es 42 ezerért, kívánságra + fél Mbájt memória + 2000,— Ft. C 64-re TAPE-TOOLS, FINAL III., FASTLOAD cartridgek. Hibás számítógépeket, tartozékokat (modulátor, tápegység) vennék. Tel.: 184-8845.

Enterprise programok eladók. Válaszboríték ellenében listát küldök. 2000 program, sok kedvezmény, ajándék. Cím: Zemen László, 1164 Budapest XVI., Áföldi Ltp. 1035. Ép. II. Lh. TT. 7.

Enterprise-osok figyelem! Szenzációs ajánlat! Aki december 31-ig egyszeri megrendelést ad fel, az egy Casio típusú telememós karórát nyerhet, valamint 10 darab több programot tartalmazó kazettát is kisorsolok. A nyereményjegyzéket a februári számban közlöm. Ugyanitt 2000 program eladó. Sok kedvezmény, ajándék. Válaszboríték ellenében listát küldök. Cím: Zemen László, 1164 Budapest XVI., Áföldi Ltp. 1035. Ép. II. Lh. TT. 7.

Enterprise-szenzáció! Aki december 31-ig 3000 forintig egyszeri programmegrendelést ad fel, az részt vesz azon a sorsoláson, amelyen egy Casio típusú telememós karórát (6900 Ft) és 10 darab programkazettát nyerhet. Figyelje a februári, márciusi számokat! Ugyanitt 2000 program eladó. Sok kedvezmény, ajándék. Válaszboríték ellenében listát küldök. Cím: Zemen László, 1164 Budapest XVI., Áföldi Ltp. 1035. Ép. II. Lh. TT. 7.

CLIPPER 87 EXE és OVL programok visszaalakítása forrás formátumúvá saját fejlesztésű programmal, 100%-os hibamentességgel. Cím: DECOMPILER STUDIO, 6001 Kecskemét, Pf.: 298. Tel.: (06-76) 22-888, telefax: (76) 21-181, telex: (76) 26-515.

IBM XT/AT játék- és felhasználói programokból óriási a választék. Cím: Szőnyi László, 1161 Budapest XVI., Tavirózsa u. 5. Tel.: 184-8471.

IBM PC programok olcsón eladók! Például: LARRY5, KQ5, SQ4, PCTOOLS 7.0, JONES, INDY 3 ADV., HERO'S QUEST 2., RED BARON, WINDOWS 3.0, stb... Cím: Marosvári Zsolt, 1122 Budapest XII., Határőr út 51. Tel.: 202-0923

IBM PC XT TURBO jutányos áron eladó! Cím: Lucz Géza, 7400 Kaposvár, Kinizsi Ltp. 5. Tel.: (06-82) 11-650.

ÖTÖD-ÖLŐ játék 300 forintért eladó. Cím: Csupor László, 1539 Budapest, Pf. 720. Tel.: 115-4352.

Star LC24-10-es, 24 tűs nyomtató eladó! Csatolható IBM PC-hez, Amigához, C-64-hez. Ugyanitt profi programozók programírást vállalnak IBM PC-re, bármilyen programnyelven. Tel.: 164-5442

Sürgősen eladó egy szinte új **286-os AT** monitorral és winchesterrel. Ugyanitt egy **386-os konfiguráció** kedvező áron eladó. Tel.: 164-5442

Kései vásárfia

SYS SYSTEMS 91

A lapzárta nagy úr.
Novemberi számunknak szükségképpen készen kellett lennie a Compfair megnyitásáig, így vásári tudósítás sem kerülhetett bele. Majd' két hónappal pedig a számítástechnikai csúcsrajáratás időszaka után — úgy véljük — már nem illendő valamiféle szabályos Compfair-, illetve Systems-beszámolóval előhozakodni. Inkább úgy döntöttünk, hogy csak néhány apró emlékképet — és termékképet — villantunk fel mindkét helyszínről.

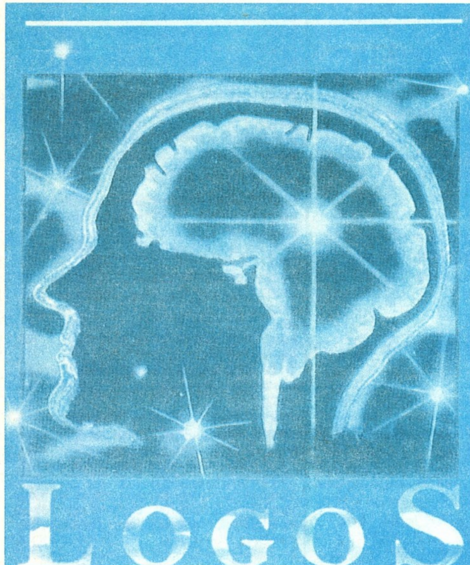


COMPFAIR 91

Egy ésszerű lépés

Mivel az IBM-kompatibilis számítógépek ismerete már a mindennapi életben is követelménnyé válik, a számítógép oktatási alkalmazásán kívül is indokolt, hogy a tanulók lehetőleg már az általános iskolában megismerkedjenek vele. Egyre több szakoktatási intézmény is érzi annak szükségét, hogy számítógépes hálózatot vásároljon. Ezzel egy időben a hazai hardverpiacon dúl az árháború.

Sajnos sok forgalmazó még mindig — az árcsökkentés érdekében — tömegcikként vásárolt egységekből, minden fejlesztői munka nélkül állítja össze konfigurációit. Ezeknek a gépeknek a megbízhatósága finoman fogalmazva is kérdéses. Mivel a forgalmazók a nagykereskedelmi árak függvényében változtatják szállítóikat, az általuk ajánlott csere-



garancia is megkérdőjelezhető, hiszen előfordulhat, hogy a garanciális idő alatt megszakítják kapcsolatukat eddigi szállítóikkal. A fenti két tendencia együtt azzal a veszéllyel jár, hogy a nehéz gazdasági helyzetben lévő oktatási intézmények és a beszűkült piacon vergődő forgalmazók találkozásával kétes minőségű számítógépek árasztják el az iskolákat.

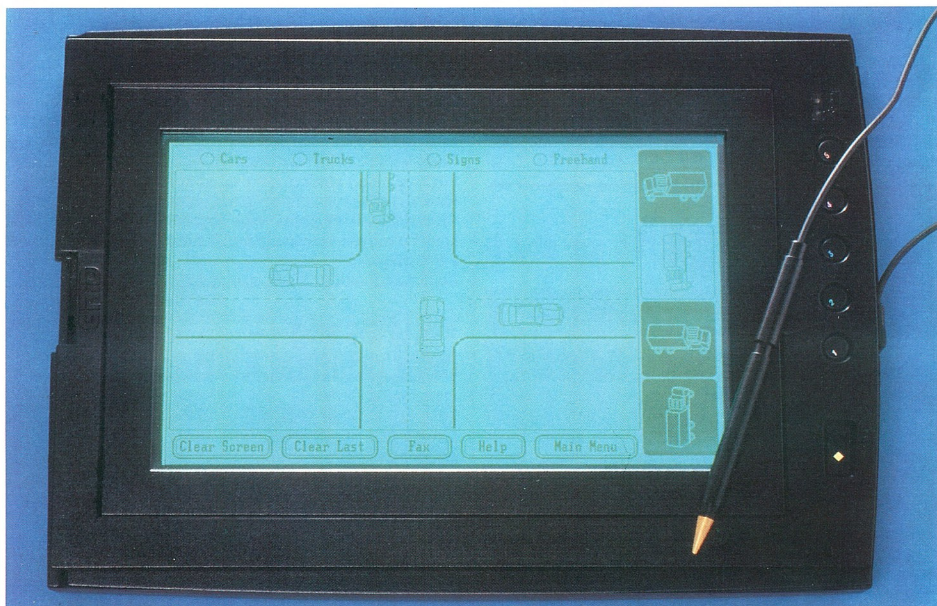
E helyzeten kíván változtatni a Logo-Press Bt., amikor a DTK számítógépek forgalmazójaként kidolgozta szolgáltatásait az oktatási intézmények számára. (A DTK gyártójának, a Datatech Enterprises Co. LTD-nek Németországban, Franciaországban, Spanyolországban, Ausztriában, az USA-ban, Japánban és Magyarországon van képviselete, számítógépeik hálózati kompatibilitását a Novell cég teszteli és igazolja. A DTK megvásárolja a legelterjedtebb rendszerek és felhasználói programok licencét.)

A DTK számítógépek árai 10-15%-kal magasabbak ugyan a nyomott árakon forgalmazott „noname" konfigurációknál, de ezért a felárért — csakúgy, mint más, jóérzésű forgalmazóknál is — a vásárló egy alapos fejlesztőmunka eredményeként kifejlesztett, nagy teljesítményű gépet kap. Ráadásul a Bt. a gépek telepítését és installálását ingyen végzi.

Hálózattelepítés esetén a cég munkatársai a helyszínen — felmérve a vásárló igényeit és lehetőségeit — alakítják ki az optimális hálózatot. Amennyiben az intézménynek az általuk szállított konfiguráció nem felel meg, egyedi gépek esetén 8, hálózat esetén 30 napon belül a teljes vételárat visszatérítik. A megvásárolt gépekre 2 év cseregaranciát adnak, illetve a tőlük vásárolt konfigurációkat a partnerek a régi egységek értékének beszámításával bővíthetik.

A Logo-Press Bt. országos szakértői hálózatot állít fel. Tapasztalat, hogy a felhasználóknál adódó hardveres problémák nagy részét a gépek helytelen setupolása, illetve a rendszerfájlok helytelen megírása okozza. A hálózat tagjai egy előre meghatározott kiszállási díj ellenében ellenőrzik a társaság által forgalmazott konfigurációkat. Amennyiben a problémát az említett hibák valamelyike okozta, a szakértő elhárítja a hibát, a díjat a felhasználó téríti meg. Ha azonban a hardver hibásodott meg, akkor a szakértő díját a forgalmazó téríti, és elvégzi a szükséges cserét. Ily módon kiküszöbölhető a felesleges kiszállásokból adódó időveszteség. A szakértők címlistáját valamennyi vásárló megkapja.

A cég az intézmények tájékozódásának megkönnyítésére egy adatbázist jelentet meg, amely tartalmazza valamennyi — a tervek szerint több ezer —, általuk forgalmazott szoftver rövid leírását. Az adatbázisban tetszés szerinti keresési módokra, összehasonlításokra nyílik lehetőség.



Mire jó a palatábla?

Az állandóan „zsibongó" Kontrax-standon sok kiállításlátogató kíváncsian nézegetett, próbálgatott, forgatott egy különös méretű és formájú „masinát". A most bemutatkozó „palatábla" a hordozható gépek kategóriájába tartozó, úgynevezett pen-computer. Az A/4-es méretű, lapos, magnéziummal borított kemény, strapabíró számítógépet az Öböl-háborúban még tankokkal sem tudták elpusztítani. Az amerikai Grid cég által fejlesztett pen-computer olyan IBM-kompatibilis számítógép, amelyen az MS-DOS operációs rendszer mellett Windows alatt is működik.

A kiállított palatáblának 2 Mbájtos memóriája mellett a 20 Mbájtos merevlemeze kivehető, és más számítógépekbe behelyezhető. Mivel a palatábla hozzákapcsolható más számítógépekhez, így az azokból származó információkat a palatábla is átveheti. A pen-computer érdekessége, hogy a kézírást is felismeri, és azt azonnal nyomtatott írássá konvertálja. A palatábla tulajdonosának kézírását — legalábbis aláírását — „megtanulja", így lehetővé válik, hogy csak „illetékes" férjen hozzá a pen-computerhez. A számítógépbe a legkülönfélébb nyomtatványokat, kérdőíveket vihetjük be, a megfelelő válaszokat pedig a palatáblához tartozó tollal egyszerűen csak kitöltjük.

Ezt a „masinát" minden olyan helyen kitűnően lehet majd használni, ahol tollal és sok-sok papírral dolgozunk. Ez azonban természetesen nem azt jelenti, hogy a palatábla akár csak egy írógépet vagy egy szövegszerkesztővel „felszerelt" számítógépet is helyettesítene. Csupán könnyű kezelhetősége miatt mindazokon a helyeken, ahol rengeteg információ érkezik be, kiválthatja a tollat és a papírt. A palatábla egyik „kistestvéréhez", a kézben is elférő handcomputerhez még vonalkódolvasó is csatlakoztatható. Így például leltározásnál, vagy vasutaknál a nagy tömegben beérkező áruk nyilvántartásánál is jól lehet használni.

Érintőleg...

A Compfair '91 egyik kedves meglepetése volt a Touch-info, amelyet hat helyszínen „nyomogathattak" a kiállításlátogatók. Látszatra „csak" egy színes monitor, valójában zárt információs rendszer és reklámmédia. Az IBM PC színes VGA monitorjára egy néhány mm vastag műanyag ernyőt helyeztek. Az érintős monitor elve alapján működő információs automata csak meghatározott nyomóerő hatására mozdul el. A 14 inches monitor 140 független pozíciót ismer fel. A 0,1 inch felbontású monitoron így nagyon jó minőségű képet láthatunk.

Az állandóan villogó színes monitoron négy nyelven — angolul, németül, olaszul és magyarul — váltakozva jelenik meg: Touch me, Tappen mich, Tocca me, Érints meg! Az első érintésre a gép lehetővé teszi a kommunikáció nyelvének kiválasztását. Nem kell mást tenni, mint a képernyőn megjelenő négy négyzet közül a kívánt nyelv feliratát megérinteni. Ezután az információs főmenü mint egy tartalomjegyzék segít az eligazodásban. Egy újabb érintés után az almenük végpontjaiban több mint ezer, konkrét névvel, címmel, emblémával ellátott, reklámmal kiegészített, nyelvileg lektorált, grafikusan illusztrált, három képernyőoldalnyi információt tanulmányozhat az érdeklődő. Egy újabb érintés hatására a kiválasztott végpontról fényképet vagy nyomtatott szöveget is kaphatunk.

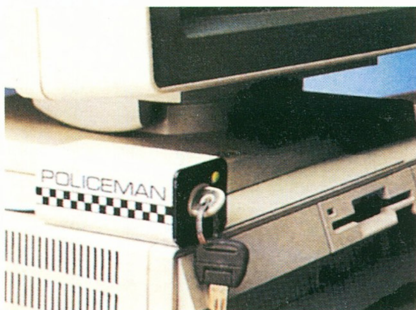
A Touch-info olyan ingyenes szolgáltatás, amelynek hatékonyságáról a hirdetőket negyedévenként ingyenesen informálják. Ugyanis a szoftver statisztikát készít, hogy az

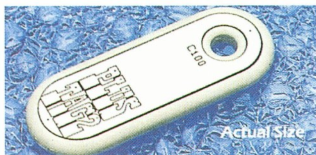
egyes végpontokat mikor és hol használták. A Touch-info népszerűsége abban is rejlik, hogy az éves hirdetési díj 50 000—70 000 Ft végpontonként, amely a mostani hirdetési díjak mellett méltányos árnak számít. A DataPlan Rt. és a Touch Info Kft. szakemberei a Compfair idejére számítástechnikai anyaggal is feltöltötték a Touch-infót. 300 cég kiállítási anyagát dolgozták fel, ezzel is segítve a látogatók jobb tájékoztatását.

Rendőrrel őriztessük számítógépünket?

Ötletes reklámfogással élt a MÜSZI MARGO a Policeman adatvédelmi rendszer bemutatásakor. Angol bobbynak beöltözött szakemberek leállították a nézelődő látogatókat, és nagy lelkesedéssel magyarázták a Policeman működését. E sorok írója „előállítása" során tudta meg, hogy a bármilyen IBM-kompatibilis számítógépen használható Policeman három lényeges alkotóeleme egy interfészkártya, egy Plus Tag adathordozó és egy olvasóazonosító egység.

A Policeman installálása után az adott számítógéphez való hozzáférést a rendszer úgy korlátozza, hogy egy 4 karakteres kódból (Personal Identification Number) és egy 8 karakteres jelszóból generál egy olyan kombinációt, amely átkódolja a floppy, a winchester, sőt a DOS tartalmát úgy, hogy az semmilyen változást nem jelent az adott számítógépen dolgozó számára. Mivel a lemez vagy a lemezegység csak a megfelelő kód és jelszó ismeretében dekódolható, így a továbbiakban csak „illetékes" férhet hozzá a számítógéphez. Az átkódolt lemezek így biztonságosan szállíthatók, hiszen azokat „illetéktelen" személyek nem olvashatják-írhatják. Sőt másik számítógépen sem használhatók ezek a lemezek, hiába is próbálják például debuggerrel visszafejteni a winchestert, az eredmény egy értelmetlen adathalmaz lesz. A Policeman a felhasználó távollétében sem enged meg illetéktelen hozzáférést a számítógéphez. Figyeli a „próbálkozásokat", és a harmadik sikertelen kísérlet után — ahogy ez egy riasztótól elvárható — szirénázik is. Ugyancsak szirénázást hallunk, ha a Policeman valamelyik alkotóelemét „kitépik".





Az angol PLUS 5 cég által fejlesztett adatvédelmi rendszer hardverúton védi meg a számítógépet az illetéktelen felhasználóktól. A hálózatban is használható, jelenleg még COCOM-listás, 49 900 forintos szoftver nagy érdeklődést váltott ki a Compfairen elsősorban azon területek szakemberei (bankok, rendőrség stb.) körében, ahol az adatbiztonság létfontosságú. Kíváncsiak vagyunk, hogy a Magyarországon a mindenfajta védelemmel — jogosan vagy jogtalanul — szemben állók körében milyen visszhangra talál a számítógépek őrangyala, a Policeman.

Vonalasság — nem politikai kérdés



Láttuk is meg nem is. Ott is volt a Compfairen meg nem is. Valahogy így jellemezhetjük azokat a vonalkódos rendszereket, amelyek jellegüknél fogva nem kiállíthatók, inkább csak egy-egy termékrészlettel, -ismertetővel jellemezhetők, ugyanakkor jelentőségük a mindennapi életben egyre nő. Különösen igaz ez például a Mag ICS Kft. vonalkódos szállítási vagy postai rendszerére, amely a munkafolyamatok valamennyi térbeli és adminisztrációs pontján hivatott biztosítani a gyors és precíz működést.

A termékeket közvetlenül a készáruraktárba érkezésükkor vonalkódos címkével látják el. A címke — amelynek előállítását ipari etikettnyomtató végzi — a szállítólevéllel analóg módon tartalmazza a küldeményre vonatkozó adatokat: fuvarlevél, szállítólevél száma, feladási és rendeltetési hely, darabszám, súly, szállítási specifikáció. Kézi adatgyűjtőkkel a küldemények ezután bármikor, bárhol azonosíthatóak. Az adatátvitel és az adatok összevetése a megbízásokkal, valamint a szállítólevelekkel lehetővé teszi a hibák azonosítását és kiküszöbölését, a számlázási rendszerben lehetőség nyílik a korrekcióra.

Sziebig Andrea

Faxolj a lemezről!

Az utóbbi évek egyik legsikeresebb technikai bravúrjáról, a telefaxról már-már azt hittük, hogy az maga a tökély, s erre megjelenik a piacon a Systems '91-en is látott DiskFax, amelynek reklámozói mindenekelőtt igyekeznek eloszlatni illúzióinkat a papírral működő faxról. Kifogásaikat 5 pontba szedték össze:

1. Papírdömping. Torlódás a küldemények adagolásánál, többszöri ismétlés az olvashatatlan sorok miatt, körlevelek küldésekor idegtépő a fax „etetése".

2. Gyenge minőség. Ha azt szeretnénk, hogy szövegünk az eredeti kivitelben érkezzen meg, akkor tegyük félre a faxot és hívjunk inkább küldöncöt.

3. Csigatempó. Frankfurtból Berlinbe egy oldal átfaxolása 51 másodperc, 3 oldalé 144, 48 oldalé 2016 másodperc.

4. Magas költség. Ha napközben és szép lassan elküldünk 10 faxot, azzal feleslegesen növeljük saját kiadásainkat.

5. Hiányos felhasználhatóság. Nem olvasható géppel, nem korrigálható, nem alkalmas nyomdai eredetinek.

A DiskFaxot forgalmazó MacByte GmbH (Aachen) ezek után a fenti 5 mínuszpont tükörképeként sorolja fel az újdonság előnyeit. Ezek mindegyike abból a technikai megoldásból következik, hogy a berendezés a dokumentumokat közvetlenül lemezről vagy a PC memóriájából továbbítja faxon, ezáltal a képek, a nagyfelbontású grafikák és rajzok is az eredetivel azonos minőségben érkeznek meg, ráadásul 20-szor gyorsabban. (A Frankfurtból Berlinbe továbbítandó fax 1 oldal esetén 2 másodperc alatt, s még a 48 oldalas is 80 másodpercen belül átmegy.)

A DiskFax berendezés 2 floppy-meghajtóval készül (5,25" és 3,5"), de kívánságra merevlemez is lehet benne. Az elküldendő dokumentumokat csak rá kell másolni egy lemezre, amit a meghajtóba helyezünk. Ha az „adás" gomb lenyomása után a telefonszámot bebillentyűztük, akkor az „indulás" gombbal már el is küldtük a faxot. Vételre állításkor üres lemezt teszünk a meghajtóba és a „vétel" gombot nyomjuk meg. Merevlemez esetén pedig az adás és a vétel közötti váltás teljesen automatikusan megtörténik.



A készülék a soros csatlakozón keresztül állhat összeköttetésben a nyomtatóval vagy a számítógéppel. ...És természetesen szükség van hozzá egy telefonvonalra is. (Rossz nyelvek szerint a magyar körülményekhez való alkalmazkodás érdekében a fejlesztők dolgoznak a telefonvonalat nem igénylő faxberendezések kifejlesztésén is!)

Iroda az aktatáskában



A Systems '91 nagy slágerei a hordozható (laptop, notebook) számítógépek voltak. Összeházasítva őket a mobil telefonokkal és még néhány tartozékkal, máris összeáll egy aktatáskában elférő, komplett mobil iroda. A Speech Design GmbH erre a feladatra specializálta magát és bemutatta a NoDesk termékcsaládot. Az elegáns diplomatatáskában olyan márkás készülékek vannak rendszerbe szervezve, amelyek révén „mindig minden kéznél van", bárhol is tartózkodunk. A táska „tartalomjegyzéke":

— Hálózati adapter
— Notebook számítógép
— CD-ROM
— Egér
— Nyomtató
— Adó-vevő modem
— Mobiltelefon
— Mobilfax

A kinyitott táska a benne lévő eszközök jó elrendezésével rögtön irodaként használható. Az összekötő kábelek rejtve vannak. Valamennyi készülék egyetlen hálózati csatlakozóról működik, de hálózattól függetlenül, elemekkel is használható.

Pesties zsargonban kommentálva: „Hát nem semmi!"

Adatbank a kompakt kártyán

Ha így halad tovább, hamarosan mellényzsebben elférő lexikonokkal sétálgathatunk. A kaliforniai Szilikon Völgyből ismét előjött valami nagyon vonzó újdonság, s azt a müncheni Systems '91 szakvásáron is bemutatták. A Drexler Technology Corporation ugyanis kifejlesztett és a LaserCard Systems Corporation forgalomba hozott egy hitelkártya méretű (8,5 x 5,5 cm-es) optikai adattárolót. A LaserCard

digitális adathordozó rétege átlátszó polikarbonát fóliák közé van hegesztve. Egy adatpont mérete 2,25 mikron (ezred mm). A kártya író-olvasó készüléke csatlakoztatható a PC-hez vagy más mikrogéphez. A technológia (WORM = write once read many times = egyszer írható, akárhányszor olvasható) hasonlít a kompakt lemezéhez, azzal a különbséggel, hogy a CD-ről csak olvashatunk, oda írni nem tudunk, a lézerkártyára felvitt adatokhoz viszont később is bármikor hozzá lehet fűzni újabb információkat. A beírt adatok azonban utána már itt sem módosíthatók és nem törölhetők.

A kártyán tulajdonképpen mindenféle digitalizálható információ elhelyezhető: szöveg, kép, beszéd, zene, számítógépi program. A normál kivitelű LaserCard tárolókapacitása 4,1 megabájt, ami az EDAC (error detection and correction) hibafelismerő és korrigáló eljárás alkalmazásával ugyan 2,9 megabájtra csökken, de még az is több, mint az Alaplapban fél év alatt közölt cikkek terjedelme együttvéve. A kártya teljes tárolókapacitása akkor használható ki, ha hibakorrekciót nem igénylő anyagokat (képeket, grafikákat) viszünk fel rá. (Pl. röntgenfelvételek, ultrahangképek stb.)

Könnyen lehet, hogy ilyen lézerkártya lesz majd az összekötő kapocs az ember és a számítógép között, hiszen szó szerint „zsebrevágható”, állandóan magunknál hordható, de borítékban is elküldhető és egyszerűen archiválható is. Alkalmazása különösen ígéretes a kórházak, hatóságok, bankok számára. Rajta lehet például teljes egészségügyi dokumentációnk, telefonkönyvünk vagy akár összes igazolványunk tartalma... A kártya teljesen érzéketlen a mágneses erőtérrel és az elektrosztatikus hatásokkal szemben, s mert adatmódosításra és adattörlésre sincs lehetőség, érthető a biztonságtechnikai szakemberek élénk érdeklődése iránta.

A kártyára rátehető tulajdonosának szemmel láthatatlan arcképe és ujjlenyomata, akár az olvasóberendezés számára is titkosítva, hogy csak az nézhesse meg, aki a kód birtokában van. Külön meghatározható az egyes adatállományokra a hozzáférés kulcsa vagy letiltása. S hogy még érdekesebb legyen: az optikai tárolóréteget a lézerrel oly módon is lehet „feliratozni”, hogy a kártyán szemmel is látható jelek, képek, emblémák jelenjenek meg. A Drexler cég felkészült a kártyák gyártására, éves kapacitásuk 25 millió darab.

Egérfogás

A legtöbb egér szürke, mint az egér. De miért ne lehetne olyan tarka, mint a pillangó? (A számítógép mellett már nincs macskaveszély!) És miért ne lehetne jobban tenyérbesimuló? A tajvani New Idea Electronic Co. ilyen megfontolásból fejlesztette ki a színes, bogárhátú, áramvonalas „egércsaládot”, elterjesztve a „két és fél gombos” változatot is, ahol a baloldali és jobboldali gomb jóval nagyobb felületet kap, mint a ritkábban használt középső. A cég innovatív szellemének — és Patrick Wu tervezőnek — legújabb különleges terméke pedig az első pillantásra egészen misztikus látványt keltő „Legend Ball” egér, amely precizitása mellett állítólag kifejezetten pihentető a kéznek — nem is szólva az erotikus képzettársítás megannyi lehetőségéről.

Faklen Pál







Ha a megbízhatóság a döntő...

Super Compact 386SX Workstation

VIGYÁZAT! Jól bevezetett és hírnévnek örvendő márkanevünkkel
kétes minőségű, hasonló hangzású nevek élnek vissza!

A MITAC 17 éves információipari hátterével a technológia egyik távol-keleti vezetője.
Igen szigorú minőségbiztosító rendszerének és hatalmas kutató-fejlesztő beruházásainak
eredményeképpen termékei a világ 65 országában váltak a korszerűség és a megbízhatóság szinonimájává.

A megbízható gyártó termékei csak megbízható forgalmazó tevékenysége nyomán képesek
a felhasználó javát szolgálni.

Ezért esett a MITAC választása hazánkban az INTERAG-ra.

Forgalmazó:



INTERAG INFORMATIKA
Budapest 1136 Pannónia u. 11.
Tel./fax.: 132-9375 Molnár Péter

People Committed To InfoTech

MiTAC



INFORMÁCIÓKÉRÉS: 05 ▶